

Quantitative Approaches in Object-Oriented Software Engineering

Fernando Brito e Abreu¹, Brian Henderson-Sellers², Mario Piattini³, Geert Poels⁴, and Houari A. Sahraoui⁵

¹ Universidade Nova de Lisboa & INESC, Rua Alves Redol, 9, Apartado 13069,
1000 Lisboa, Portugal

fba@di.fct.unl.pt – fba@inesc.pt

² University of Technology Sidney, Faculty of Information Technology, PO Box 123,
Broadway NSW 2007, Australia

brian@it.uts.edu.au

³ Universidad de Castilla-La Mancha, Ronda de Calatrava, 5, 13071, Ciudad Real, Spain
mpiattini@inf-cr.uclm.es

⁴ Katholieke Universiteit Leuven & VLEKHO Business School, Koningsstraat 336,
1030 Brussels, Belgium

geert.poels@econ.kuleuven.ac.be – gpoels@vlekho.wenk.be

⁵ Université de Montréal, CP 6128 succ Centre-Ville, Montréal QC H3C 3J7, Québec, Canada
sahraouh@iro.umontreal.ca

Abstract. This report summarizes the contributions and debates of the 5th International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2001), which was held in Budapest on 18–19 June, 2001. The objective of the QAOOSE workshop series is to present, discuss and encourage the use of quantitative methods in object-oriented software engineering research and practice. This year’s workshop included the presentation of eight position papers and one tutorial in the areas of “software metrics definition”, “software size, complexity and quality assessment”, and “software quality prediction models”. The discussion sessions focused on current problems and future research directions in QAOOSE.

1 Introduction

The 5th International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2001) was a direct continuation of four successful workshops, held at previous editions of ECOOP in Cannes (2000), Lisbon (1999), Brussels (1998) and Aarhus (1995).¹ Like in previous years, the workshop attracted participants from academia and industry that are involved or interested in the application of quantitative methods in object oriented (OO) software engineering research and practice. A list of this year’s workshop participants, including their contact information, can be found at the end of this report.

¹ The main objectives, themes, and results of the QAOOSE’99 and QAOOSE 2000 workshops are summarized in reports that have been published in Springer-Verlag LNCS ECOOP workshop readers [1,2].

Quantitative approaches in the OO field is a broad but active research area that aims at the development and evaluation of methods, techniques, tools and practical guidelines to improve the quality of software products and the efficiency and effectiveness of software processes. The relevant research topics are diverse, but always include a strong focus on applying a scientific methodology based on data collection (either by objective measurements or subjective assessments) and analysis (by statistical or artificial intelligence techniques). Like in previous years, submissions of position papers were invited, but not limited, to the areas of metrics collection, quality assessment, metrics validation, and process management.

The 2001 edition of the workshop aimed to shed some light on recent research results and to point out future research directions that might interest not only the academic community but also industry. Over the years the QAOOSE workshop series has built an active research community working towards special topics of interest that were identified as open issues during previous workshop editions. This year we explicitly solicited contributions of new research on (i) automatic support to share research hypotheses, data, and results, (ii) measures for non-functional requirements, (iii) measures for component models, (iv) meta-metrics, and (v) the application of quantitative methods in industrial software processes.

A concrete result of the workshop series is a forthcoming Addison-Wesley book (to appear in 2002), which will contain chapters authored by participants selected from all previous workshop editions. Another product of QAOOSE is a special issue of the journal *L'Objet: Software, Databases, Networks* on quantitative approaches in OO software design [3]. This issue includes six externally refereed papers which are thoroughly extended and revised versions of position papers presented at the QAOOSE 2000 workshop.

This report is organized as follows. In Sect. 2 we present a comparative summary of this year's workshop contributions, organized by workshop session and topic area. In section 3 a summary of the workshop debates is given. In section 4 we present the workshop conclusions and the open issues that were identified. This report ends with the complete list of contributions,² the contact data of the workshop participants, and some relevant references.

2 Summary of the Contributions

Usually, the application of quantitative methods in software engineering follows a logical path (i) from the construction or selection of measurement instruments, (ii) over the measurement-based assessment and evaluation of software product quality, software process efficiency, and software engineering productivity, (iii) to the development and use of prediction models, based on historical measurement data.

On June 18, our workshop followed this structure. The position papers that were presented this day were organized into three presentation sessions, called (i) Metrics

² All contributions accepted to the workshop were published in the workshop's proceedings [4]. To obtain a copy of the proceedings (ISBN 90-806472-1-7), contact the executive editor (Geert Poels – gpoels@vlekho.wenk.be). Electronic versions of the position papers and tutorial materials (all in PDF format) can also be downloaded from the workshop's web pages (URL: http://www.iro.umontreal.ca/~sahraouh/qaoose01/QAOOSE01_program.html).

Definition, (ii) Assessment Models, and (iii) Predictive Models. On June 19, the Metrics Definition session, and actually the workshop itself, was continued by a tutorial given by Fernando Brito e Abreu on the definition of OO software metrics using the Object Constraint Language (OCL). It was the first time that a QAOOSE workshop was extended to a one-and-a-half-day event.

During the presentation sessions each attendee was asked to summarize the new ideas and limitations of each position paper presented. Our experience with last year's workshop (QAOOSE 2000 in Cannes) learned us that this exercise greatly helped to stimulate the workshop discussions. That is why we decided to continue this tradition.

It was recognized that the distribution of position papers across the three sessions, each meant to cover a different topic area, was at times arbitrary. Therefore, and also in order to facilitate the comparison between the contributions, we decided to split this section into just two sub-sections. In a first sub-section we present, discuss, and compare the contributions that deal with meta-models for metrics definition, the proposal of new metrics suites, or any other proposal related to software measurement or assessment instruments. In a second sub-section we do the same for contributions that focus on the use of metrics in descriptive or predictive models, or the actual development of these models. The latter sub-section also includes contributions presenting an empirical validation of proposed metrics.

2.1 Metrics Definition

Many of this year's workshop contributions relate to the definition of new metrics suites or other types of assessment instruments, or include proposals to improve the state-of-the-art in OO software measurement and quality assurance.

Problems with the currently available measurement instruments for OO software artifacts have been identified and discussed in previous QAOOSE workshops (see [1,2]). One problem with many, if not most, of the OO metrics suites is their lack of precision in metrics definitions. The resulting ambiguity is a severe hindrance to the widespread acceptance and use of metrics (and quantitative approaches in general) in industry. Without clear and precise definitions, it is impossible to examine the theoretical validity of the metrics, to replicate experiments aimed at the empirical validation of the metrics, and to automate the metrics collection process. To guarantee precision in metrics definitions a formal language is needed. The language for the metrics definitions should be (consistent with) the language in terms of which the entities to be measured, their interrelationships, and their constraints are specified. Clearly, current metrics suites have not been defined with respect to a formal meta-model of their object(s) of measurement, i.e. the type(s) of OO software artifact for which they are intended.

Interestingly, this year's workshop participants witnessed the presentation of two related, but distinct proposals for metrics definition based on a formal meta-model of OO design. In one of the sessions, Ralf Reissing introduced his meta-model called Object-oriented DEsign Model (ODEM). ODEM is based on the UML meta-model, which is the "de facto" standard meta-model for OO design. As ODEM is intended for the definition of metrics for high-level design, it borrows only those parts of the UML meta-model that relate to class and package diagrams. However, as explained by Reissing, the model can easily be extended to include other model elements (e.g.

object behavior). The added value of ODEM is a layer of abstractions defined on top of the UML meta-model. This new layer contains formally defined sets, relationships and attributes in terms of which metric definitions can be stated. Reissing further delivered a proof of concept by defining some example class, package and system metrics, and by redefining existing metrics (all of Martin's Package metrics and some of Chidamber and Kemerer's metrics) in the language of ODEM.

The competing proposal came from one of the workshop organizers, Fernando Brito e Abreu, in his tutorial on the second day of the workshop. There are remarkable similarities in both works. The research motives are nearly the same, and both meta-model proposals use elements of the UML meta-model. However, the approaches taken by these researchers to define metrics are different. Brito e Abreu does not define an additional layer on top of the UML meta-model, but instead uses OCL to define metrics as meta-model operations. The applicability limitations of the metrics can then be defined with OCL preconditions; the metrics results with OCL postconditions.

An advantage of Brito e Abreu's approach, compared to Reissing's proposal, is the greater standardization that comes along with it. OCL, as part of UML, is becoming very fast a standard specification language for invariants and assertions in OO design. A disadvantage might be that it requires the user to learn and master OCL,³ whereas the set theoretic concepts in ODEM seem to be more accessible for non-programmers and non-modelers. Interestingly, in his position paper, Reissing explicitly motivates his choice not to use OCL (see p. 75 in [4]): "It was tried to use the OCL as a specification language for design metrics, but it does not seem to be suited for this purpose. Some metrics were extremely hard to express in OCL, and most of the resulting metric definitions were hard to read".

Another advantage of Brito e Abreu's approach is its flexibility. In his technical report, accompanying the tutorial, Brito e Abreu first describes (and formalizes) the GOODLY (Generic Object Oriented Design Language? Yes!) language, previously used to define the MOOD metrics, using UML and OCL. Next, the definition of the metrics in his new version of the MOOD suite, i.e. MOOD2, is specified using OCL. The approach of Brito e Abreu is however not limited to GOODLY, or even UML. In his outlook to further work, he describes ongoing and planned work where metrics will be specified using OCL directly for the UML meta-model and for other emerging meta-models like the OPEN Modeling Language (OML).

Whereas the contributions of Ralf Reissing and Fernando Brito e Abreu mainly concerned the formalization of existing metrics suites, other contributions presented new metrics proposals. Michel Dao, Therese Libourel, and Cyril Roume described their work on a new metrics suite for the evaluation of factorization and generalization in class hierarchies. In total, 18 new metrics are proposed for four levels of granularity in OO software, i.e. features, generic features (which are sets of features), classes, and class

³ A document containing the Object Constraint Language specification can be downloaded from the workshop's web pages. It is part of the tutorial materials on metrics definition using OCL (URL: http://www.iro.umontreal.ca/~sahraouh/qaoose01/QA00SE01_program.html).

hierarchies. The purpose of these metrics is to assess how much feature factorization⁴ there is in a hierarchy of classes. The goal of the work is to incorporate these metrics in the class hierarchy construction tools that these researchers are currently building. The main principle underlying such tools is that class hierarchies must be maximally factorized, while introducing as few classes as possible for feature factorization. The proposed metrics can be used to highlight design defects with respect to feature factorization, and help software engineers chose the most promising design amongst alternative solutions.

Another metrics suite proposal was done by Marcela Genero. She briefly presented a set of 10 structural complexity metrics for UML class diagrams. The bulk of her presentation however focused on validating these metrics as maintainability indicators (cf. Sect. 2.2).

In the position paper of Ram and Raju a new metric, called Method Points, is presented to estimate the size of a software design that is composed from design patterns. This metric is especially useful in the presence of alternative designs that result from the use of different patterns. Method Points are a relative measure of size, meaning that the common parts in alternative designs are not taken into account. As a consequence, an evaluation of alternative designs based on size can be carried out with minimal effort.

Although one of the previous QAOOSE workshops has questioned the proposal of new metrics suites [1], the contributions presented and discussed in this year's workshop show that there are still unfilled niches for research on OO metrics. One promising area of research is that of metrics for database schemata.

The contribution of Olaf Herden touches upon this subject. Herden presented a framework for conducting quality reviews of database schemata. This framework consists of a formal (UML) meta-model capturing the different elements of quality reviews, and a process model for working with this meta-model. Herden further instantiated the meta-model for quality criteria of conceptual OO database schemata. He also sketched the architecture of the tool infrastructure that is needed to automate and document the reviews. Part of this architecture has been realized.

The final contribution we wish to mention in this sub-section is the one of Hind Kabaili. She and her colleagues have extended a previous change impact model to take into account the ripple effect and regression testing. The model identifies the classes in a C++ system that are affected (i.e. they need correction) by a change in another class the system. The impacted classes depend on the type of change and their relationship with the changed class. The ripple effect extension takes indirect relationships between classes into account. The model that accommodates regression testing determines which classes to retest (even if they do not need correction).

2.2 Metrics Use

Five contributions dealt with what we categorized as the use of OO metrics. There was a considerable diversity in the metrics applications presented, going from typical metrics

⁴ Feature factorization in a class hierarchy is maximal if there is only one class that declares the feature (for a formal definition see p. 86 in [4]). The technique of feature factorization is used in a generalization process that aims at (re-)designing OO software such that reusability and extensibility are improved.

validation studies, where a relationship with a variable of interest (e.g. maintainability) is established, to the actual building of descriptive or predictive models. Regarding the latter aspect, it is worth mentioning that some contributions used techniques from the Artificial Intelligence and Machine Learning fields; a novel approach that is observed across the whole software measurement and empirical software engineering field. One of the QAOOSE workshops organizers, Houari A. Sahraoui, is a mayor proponent of the use of Machine Learning techniques in software quality models, and his influence and example is strongly felt in the QAOOSE research community.

In one of the sessions, Marcela Genero presented a typical metrics validation study.⁵ She reported on a controlled experiment that was set up to evaluate the correlation between 10 UML class diagram structural complexity metrics (cf. Sect. 2.1) and subjective ratings of three UML class diagram maintainability sub-characteristics, i.e. their understandability, analyzability, and modifiability. The subjects were 7 professors of the Department of Computer Science at the University of Castilla-La Mancha (Ciudad Real, Spain) and 10 students enrolled in the final year of Computer Science. All subjects had to rate the understandability, analyzability, and modifiability of 28 class diagrams using a 7-point Likert scale, ranging from "Extremely difficult to understand (analyze, modify)" to "Extremely easy to understand (analyze, modify)". Spearman's correlation coefficient was used to analyze the correlation between these ratings and the values of the 10 structural complexity metrics. The analysis showed that there exists a significant correlation between most of the metrics and the maintainability ratings, which provides empirical evidence of the hypothesized relationship.

The discussion of Genero's work focused mainly on fundamental questions such as the evaluation of a design representation (i.e. a UML class diagram) instead of the system itself, and the investigation of class diagram maintainability instead of software maintainability. It was noted that the ultimate goal of many metrics is to use them as early quality indicators, which is why the focus of much of the research in this field is on the measurement of the internal properties of OO designs (and recently also OO conceptual models). It was however acknowledged that to obtain really useful indicators, we must also establish a relationship with the external quality of the software system that is based on the design or conceptual model.

The issue of metrics validation has also been touched upon in the presentation of Hind Kabaili. She described a future experiment that will investigate the ability of well-known OO design metrics (e.g. Chidamber and Kemerer's metrics, the MOOD set of metrics) to be used as changeability indicators for OO software. This experiment will use the extended change impact model for C++ systems, developed by Kabaili and colleagues (cf. Sect. 2.1).

In two of the workshop's presentations, the development of metrics-based quality prediction models was discussed. Marcela Genero presented and explained a new technique, called Fuzzy Prototypical Knowledge Discovery (FPKD), that extends the traditional Knowledge Discovery in Databases (KDD) process with concepts derived from fuzzy set theory. This technique was used to build a model that can predict the under-

⁵ A more complete validation, containing both theoretical and empirical parts, can be found in the Ph.D. dissertation of Marcela Genero [5].

standability, analyzability, and modifiability ratings of a UML class diagram based on its structural complexity metrics values.

The presentation of Houari Sahraoui focused on two problems with current methods to build software quality prediction models: the use and interpretation of precise threshold values for metrics, and the lack of decision support offered by "naive" models that do not explain the observed relationship between internal software properties and external software quality characteristics. The proposed solution of Sahraoui and colleagues is to map crisp threshold values into fuzzy ones, and to explicitly include domain knowledge into a prediction model. The former is done using a fuzzification technique based on the distribution of the metric values. Any metric value that is an input to a predictive model can then be replaced by the fuzzy label assigned to it (e.g. "small", "medium", "large"). For instance, if the predictive model consists of a set of rules, then the crisp preconditions of these rules can be replaced by fuzzy ones. Regarding the rules derivation itself, Sahraoui and colleagues propose breaking up rules that do not explain causality very well, into a set of intuitive and easier to validate sub-rules. This can be accomplished by including domain knowledge (e.g. domain heuristics) into the model.

The discussion of this highly original proposal focused mainly on the elaboration of the causal model. It was acknowledged that the derivation of the sub-rules, which formalize relationships between intermediate causes, is hard to automate, and their justification must be done on intuitive grounds. The existence of domain-specific public repositories of validated relationships between software attributes would be of great help to the model builder. It must be noted that the creation of such repositories was identified as an open issue during the QAOOSE 2000 workshop (see [2]). Again, the urgent need to share research hypotheses, data, and results, preferably with automatic support of some kind, was stressed.

To end this sub-section we report on two contributions that propose the use of metrics for evaluation purposes. In the position paper of Poels, Dedene, and Viaene, some simple complexity metrics for existence dependency graphs, i.e. a specific type of OO static conceptual model, have been applied to five reference models for organizing a front-office. The results of this exercise have been used to determine an optimal level of service customization. In their presentation, Michel Dao, Therese Libourel, and Cyril Roume have described some ongoing experiments related to the application of their feature factorization metrics (cf. Sect. 2.1) to several JAVA packages. Measurements have been taken at four levels of granularity (i.e. feature, generic feature, class, hierarchy) to assess how well these packages were factorized. The findings of these experiments confirmed many of the already sustained factorization problems in the packages under study.

3 Summary of the Debates

The QAOOSE 2001 workshop featured a separate discussion session at the end of the first day, and a shorter one on day two to close the workshop. We do not repeat the comments here that pertain to individual or related position papers presented during the workshop. These comments are included in the text of the previous section. We focus here on a couple of general remarks by the workshop attendees, that hold for all research presented, and in fact for the QAOOSE field as a whole.

It was noticed that there are problems with cohesion metrics for OO software artifacts. Most of the published cohesion metrics, like Chidamber and Kemerer's LCOM, lack a precise definition. As a consequence, it is not clear how to compute them. Interestingly, a same comment was made in previous QAOOSE workshops (see [1,2]). Hopefully, the metrics (definition) standardization proposals made during this workshop (see Sect. 2.1) can alleviate some of the problems with OO cohesion metrics. But, as shown by Hind Kabaili in last year's workshop, semantic aspects must also be taken into account for an effective assessment of cohesion. Software metrics are syntax-based and are probably not sufficient to evaluate this property. Michel Dao and colleagues reached about the same conclusion for their factorization metrics. In general, much of this year's workshop discussions related to issues of metrics definition like meta-metrics, meta-models for metrics, and the semantic aspects involved.

Another general remark is the lack of theoretical validation of metrics for OO software artifacts. Also this area of research needs further attention. We need to make sure that it is known what is measured by a metric. We cannot expect a wide adoption of metrics-based evaluation and prediction instruments in industry if this problem remains unsolved.

Related to the previous comment is the question what data analysis techniques are best to build descriptive and predictive software quality and effort models. A comparative analysis of different techniques, including Machine Learning techniques, is very much desired. Bayesian networks of several models might also provide part of the answer.

Once there is sufficient confidence in the validity of measurement instruments and in the usefulness and effectiveness of quality or effort models based on these instruments, there is the question how to integrate all of this in the software development, maintenance, and reengineering processes. Further research in this domain is wanted too.

Another question is whether metrics can be used to validate claims about design patterns. We would for instance like to see studies that investigate whether the use of patterns actually improves the quality of software systems. A quantitative approach to studying this research question seems appropriate.

Finally, the need for metrics for dynamic design diagrams and for OO patterns, components, and architectures was reiterated, as was the plea for more public repositories of software engineering data, metric values, thresholds, and benchmarks. Regarding this latter aspect, an international effort is required.

4 Conclusions

To conclude we can safely state that QAOOSE 2001 was a success. We saw some very innovative proposals and work-in-process. Especially, the research efforts towards metrics standardization (i.e. the topic of meta-metrics) and the omnipresent focus on measuring early OO artifacts, like UML class diagrams and OO conceptual database schemata, were very encouraging. Moreover, these topics were identified as open issues during previous editions of our workshop, which is somehow a proof of the significance and impact of the QAOOSE workshop series.

We, as workshop organizers, commit ourselves to continue this workshop series and improve it further. The issues raised in Sect. 3, offer food for thought for the coming

QAOOSE workshop edition(s). We invite researchers, both established researchers and Master or Ph.D. students, to work towards these topics, and share their findings with us on the next occasion.

5 List of Contributions

The following position papers were presented at the workshop:

- Extending Software Quality Predictive Models Using Domain Knowledge by Houari Sahraoui, Mohamed Adel Serhani, and Mounir Boukadoum
 - Empirical Validation of Measures for Class Diagram Structural Complexity through Controlled Experiments by Marcela Genero and Mario Piattini
 - A Change Impact Model Encompassing Ripple Effect and Regression Testing by Hind Kabaili, Rudolf K. Keller, and F. Lustman
 - Measuring Quality of Database Schemas by Reviewing Concept, Criteria and Tools by Olaf Herden
 - Towards a Model for Object-Oriented Design Measurement by Ralf Reissing
 - Towards a Metrics Suite for Evaluating Factorization and Generalization in Class Hierarchies by M. Dao, M. Huchard, H. Leblanc, T. Libourel, and C. Roume
- Two position papers were accepted, but not presented at the workshop:
- Estimating Relative Size When Alternative Designs Exist by D. Janaki Ram and S.V.G.K. Raju
 - A Quantitative Assessment of the Complexity of Static Conceptual Schemata for Reference Types of Front-Office by Geert Poels, Guido Dedene, and Stijn Viaene
- The tutorial was documented by a technical report:
- Using OCL to Formalize Object Oriented Metrics Definitions by Fernando Brito e Abreu

6 List of Participants

The workshop gathered 14 people from 6 different countries. Their names, affiliations and e-mail addresses are:

- Brito e Abreu, Fernando (UNL & INESC, Portugal)
fba@inesc.pt
- Dao, Michel (France Télécom R&D, France)
michel.dao@francetelecom.com
- Genero, Marcela (University of Castilla-La Mancha, Spain)
mgenero@inf-cr.uclm.es
- Herden, Olaf (OFFIS, Germany)
olaf.herden@offis.de
- Josset, François Xavier (BOUYGUES, France)
fxjosset@bouygues.com
- Kabaili, Hind (University of Montreal, Canada)
kabaili@iro.umontreal.ca

- Lahire, Philippe (University of Nice, France)
philippe.lahire@unice.fr
- Libourel, Therese (LIRMM, France)
libourel@lirmm.fr
- Piattini, Mario (University of Castilla-La Mancha, Spain)
mpiattini@inf-cr.uclm.es
- Poels, Geert (VLEKHO Business School, Belgium)
gpoels@vlekho.wenk.be
- Reissing, Ralf (University of Stuttgart, Germany)
reissing@informatik.uni-stuttgart.de
- Roume, Cyril (LIRMM, France)
roume@lirmm.fr
- Sahraoui, Houari (University of Montreal, Canada)
sahraouh@iro.umontreal.ca
- Van Belle, Werner (Vrije Universiteit Brussel, Belgium)
werner.van.belle@vub.ac.be

Acknowledgments. The authors wish to thank one of the workshop participants, Marcela Genero (University of Castilla-La Mancha), for her great help in compiling and reviewing this report.

References

1. Brito e Abreu, F., Zuse, H., Sahraoui, H.A., Melo, W.L.: Quantitative Approaches in Object-Oriented Software Engineering. In: Moreira, A.M.D., Demeyer, S. (eds.): Object-Oriented Technology, ECOOP'99 Workshop Reader, ECOOP'99 Workshops, Panels, and Posters, Lisbon, Portugal, June 14-18, 1999, Proceedings. Lecture Notes in Computer Science, Vol. 1743, Springer-Verlag, Berlin (1999) 326-337
2. Brito e Abreu, F., Poels, G., Sahraoui, H.A., Zuse, H.: Quantitative Approaches in Object-Oriented Software Engineering. In: Malenfant, J., Moisan, S., Moreira, A.M.D. (eds.): Object-Oriented Technology, ECOOP 2000 Workshop Reader, ECOOP 2000 Workshops, Panels, and Posters, Sophia Antipolis and Cannes, France, June 12-16, 2000, Proceedings. Lecture Notes in Computer Science, Vol. 1964, Springer-Verlag, Berlin (2000) 93-103
3. Brito e Abreu, F., Poels, G., Sahraoui, H.A., Zuse, H. (eds.): Quantitative Approaches in Object-Oriented Software Design. Special issue of L'Objet: Software, Databases, Networks 7 (2001)
4. Brito e Abreu, F., Henderson-Sellers, B., Piattini, M., Poels, G., Sahraoui, H.A. (eds.): Proceedings of the 5th International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering. Leuven (2001)
5. Genero, M., Defining and Validating Metrics for Conceptual Models. Ph.D. dissertation, Dept. of Computer Science, University of Castilla-La Mancha. Ciudad Real (2001).