Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2005)

WEREGLA[1]

[1] Collective name for QAOOSE 2005 participants and organizers
Organizer's contact information is in this report

Abstract. The QAOOSE'2005 workshop brought together, for a full day, researchers working on several aspects related to quantitative evaluation of software artifacts developed with the object-oriented paradigm and related technologies. Ideas and experiences were shared and discussed. This report includes a summary of the technical presentations and subsequent discussions raised by them. Nine out of fourteen submitted position papers were presented, covering different aspects such as metrics and code analysis, quality assessment, and empirical studies. In the closing session the participants were able to discuss open issues and challenges arising from researching in this area, as well as they tried to forecast which will be the hot topics for research in the short to medium term.

1 Historical background and motivations

QAOOSE 2005 is a direct continuation of eight successful workshops, held at previous editions of ECOOP in Oslo (2004), Darmstadt (2003), Malaga (2002), Budapest (2001), Cannes (2000), Lisbon (1999), Brussels (1998) and Aarhus (1995). The QAOOSE series of workshops has attracted participants from both academia and industry that are involved / interested in the application of quantitative methods in object oriented software engineering research and practice. Quantitative approaches in the OO field is a broad but active research area that aims at the development and/or evaluation of methods, techniques, tools and practical guidelines to improve the quality of software products and the efficiency and effectiveness of software processes. The workshop is open to other technologies related to OO such as component-based systems (CBS), web-based systems (WBS) and agent-based systems (ABS).
The relevant research topics are diverse, but include a strong focus on applying empirical software engineering techniques.

2 Workshop overview

Fifteen (14) people, out of forty-one (41) overall participants (if we include the co-authors that were not present) attended the workshop. They were representing seventeen (17) different organizations from ten (10) different countries. Among the attendants, two (2) people were not authors, as it is normally the case in these kind of closed[1] workshops. They have asked the organizers to attend the workshop, which is an additional evidence of the interest raised by this area.

This workshop encompassed four 90-minute sessions. The topics of each session were, respectively: (1) code analysis and metrics, (2) quality assessment, (3) Empirical studies, and (4) open issues and promising research avenues. For each of the three first sessions three presentations took place. In addition, some participants were allowed a time slot for expressing their position statement. Each presentation, plus corresponding discussion, took around 30 minutes (10 minutes for position statements). Those presentations were based on submitted position papers that went through an evaluation process conducted by the organizers. In the final session, a collective effort was performed to discuss some open issues that rose from the three previous sessions and to identify future trends for this workshop.

In the next three sections (one per session), we will present for each discussed paper an abstract and a summary of the consequent discussion. Section 6 summarizes the general discussion that took place in the final session. Addition information about the workshop is given in section 7.

Full texts of all accepted position papers are available at
http://www.iro.umontreal.ca/~sahraouh/qaoose2005/QAOOSE05_pgm.htm

3 Code analyses and metrics

This session was chaired by Coral Calero. In addition to the three presentations, one position statement was discussed.

Quantitative Techniques for the Assessment of Correspondence between UML Designs and Implementations

D.J.A. van Opzeeland, C.F.J. Lange (presenter), M.R.V. Chaudron, Technische Universiteit Eindhoven

In this first presentation given by Christian, approaches to assess the correspondence between a software design and its implementation are discussed. Correspondence is important for understanding the system since designs are easier to comprehend than large pieces of source code. To assess the correspondence of a system, entities from the design are matched to pieces of source code. A matching is defined based on classifiers. Several approaches are discussed to establish such a matching: matching based on classifier names, matching based on metric profiles and matching based on structural properties of classifiers. Once this matching is defined, it is possible to detect

[1] - Participation in ECOOP workshops is usually guaranteed through the submission of a position paper.

and visualize the actual differences between design entities and parts of source code. The approaches have been validated through an industrial case study.

One participant points out the difficulty of establishing non 1-to-1 matching. Another interesting comment that was discussed is the absence of semantics in the proposed approach. The authors said that they are conscious that semantics can help improving the matching tasks.

Language Independent Metric Support towards Refactoring Inference

Y, Crespo, C, López, E, Manso and R. Marticorena (presenter), University of Valladolid & University of Burgos

Raúl presented an approach that helps to determine when and where we must refactor instead of using the programmer intuition and experience. He argued that from the bad smell concept (subjective part), and using metrics (objective part), it is possible to detect refactoring opportunities. He and his co-authors applied this approach (use of metrics in bad smells detection) through an exploratory case study. This was done independently from the program implementation language. To better support this independence, he briefly described a framework for collecting metrics that allow to reuse collection code on a wide family of object-oriented languages.

Some of the participants warned the presenter on the risk of automating the refactoring based on the values of the metrics. The answer was that they use the metrics only to find refactoring candidate classes. Following the same idea a participant asked about the guarantees that if a detected class is refactored, its quality will be improved. Raúl proposed that the refactoring can be simulated and the detection procedure reapplied to see if the initial bed smells are removed.

Another issue that was discussed following this presentation is the language independence. Indeed, the fact that a metamodel allows to represent a concept (inheritance for example) that is implemented in deferent languages is not a guarantee of independence. The different implementations can have different semantics (inheritance semantics for example). Raúl explained that they manage the independence by having a core metamodel and an extension for each considered language.

Visualisation and Analysis of Software Quantitative Data,

G.Langelier, H.A. Sahraoui (presenter), and P. Poulin, Université de Montréal

Houari presents an approach for complex software analysis based on visualization. This work is motivated by the fact that many phenomena related to software, such as its evolution and its reliability are complex and there is very little theory explaining them. Today, we have a unique opportunity to empirically study these phenomena, thanks to large sets of software data available through open-source programs and open repositories. Houari and his co-authors claim that hybrid techniques that combine automatic analysis with human expertise through visualization are excellent tools for such studies. In this context, the problem of size is circumvented by exploiting perception capabilities of the human visual system.

In the discussion, a participant pointed the problem of the difficulty of using visualization tools when facing complex analysis tasks specially when not formal documentation is provided. Houari argued that the approach and the tool are designed for two different types of users. The first user (call her analyst) is responsible for customizing the environment for a specific task. This consists in elaborating a hypothesis, associating metrics to visual attributes, validating the accuracy of the analysis environment on a set of representative programs. The second user (call him final user) perform the specific analysis task on a particular program using the customized environment.

On the Impact of Aspect-Oriented Programming on Object-Oriented Metrics,

J-Y Guyomarc'h and Y-G Guéhéneuc (presenter), Université de Montréal

This position was presented by Yann. Yann and Jean-Yves claim that a study is needed to assess the impact of aspect-oriented programming on object-oriented metrics. The goal is to see if the benefits that are supposed to be brought by the AOP paradigm are not in contradiction with the OO quality principle that can be reflected with metrics. To this end, Yann proposed the principle of an approach for conducting such a study.

The discussion that took place after this presentation concerned two topics: (1) the meaning of measuring aspect oriented programs vs. measuring object-oriented programs produced by the weaving, (2) the lack of AOP programs for conducting empirical studies to validate the claims of the AOP community.

4 Quality assessment

This session was chaired by Houari Sahraoui. It contains three presentations.

Usability Indicators for Software Components

M.F. Bertoa (presenter), A. Vallecillo, Universidad de Málaga

One of the most critical processes of Component-based Software Development (CBSD) is the selection of the set of components (either from in-house or from external repositories) that fulfil the appropriate architectural and user-defined requirements. In this context, Manuel presented a set of measures and indicators to assess one quality characteristic, the Usability, of great importance to any software product, and described the method followed to obtain and validate them. The goal is to provide an objective method that helps developers evaluate the components.
As the code of components is by definition not available, the approach considers mainly documentation metrics.

This fact brings an interesting discussion on how to measure the documentation and how to use the obtained measures. First, concerning the measurement automation, Manuel explained that the used documentation is in electronic format which ease the measure extraction. Following this explanation, another participant mentioned the well-known problem of measuring figures (not considered in this work). Another comments concerned the preprocessing of the extracted raw data. Manuel clarifies the fact that the data was grouped and not used directly.

Assessing Aspect-Oriented Artifacts: Towards a Tool-Supported Quantitative Method

E. Figueiredo (presenter), A. Garcia, C. Sant'Anna, U. Kulesza, C. Lucena, PUC-Rio & Lancaster University

Eduardo reported the efforts of his team in the ongoing development of a systematic approach to support the quantitative assessment of aspect-oriented artifacts generated through the system design and implementation. The approach is organized in a step-wise fashion and is founded on a metrics suite and a comprehensive set of complementary rules. It is supported by a prototype measurement tool and has been applied to four medium-sized software systems in different domains and with distinct degrees of complexity. This work is motivated by the fact that inappropriate use of aspect-oriented abstractions and mechanisms potentially leads to the violation of important design principles, such as low coupling, high cohesion, incomplete modularization of crosscutting concerns into aspects, and so forth. These problems are not easily detectable and an *ad hoc* analysis of large designs and implementations is often expensive and time-consuming.

Following this presentation, a participant ask for clarification about how the control of rule triggering. Eduardo explained that in their approach, they use priorities to solve conflicts when more than one rule can be executed. Other participants questioned the validity of the assessment rules and specially the definition of the correspondent threshold values. Eduardo admitted that this they are working on this issue.

Open Issues with Software Quality Models

K. Khosravi Y.-G. Guéhéneuc (Yann), Université de Montréal

Yann presented a position on issues related to existing software metrics and quality models. To circumvent these problems, he proposed an approach based on design-patterns for quality assessment. The motivation of this work is inspired by the work of the authors on measuring software quality using design patterns.

This presentation gave the occasion to discuss a well known issue of the contribution of design pattern to software quality. Indeed, some participant argued that it is far from obvious that design patterns are good for quality. Yann argued that his hypothesis is based on the GOF claims and on the conclusions of some empirical studies.

Another problem revealed by a participant concerned the inadequacy of the levels of granularity of the studied artifacts (design patterns) on one hand and the used metrics (class metrics) on the other hand. Yann explained that design patterns help incorporating knowledge about design architecture. However, the quality is studied at the class level.

The global quality evaluation by combination of characteristic evaluations was the second important topic discussed. Although the authors plan to develop a technique for aggregating partial evaluations, they didn't consider the problem yet.

Finally, the participants agreed that an important requirement to help researchers exchanging data and expertise is to define a common vocabulary in the domain of quality assessment.

5 Empirical studies

This session was chaired by Yann-Gaël Guéhéneuc. Three presentations and one position statement were discussed.

Investigating the Nesting Level of Composite State in UML Statechart Diagrams

José Cruz-Lemus (presenter), Marcela Genero, Mario Piattini, and Ambrosio Tova, University of Castilla – La Mancha & University of Murcia

José presented a study of the relation between the degree of nesting in composite states and the understandability of the enclosing UML statechart diagrams. He and his colleague attempt at providing an answer to the research question: ``Does the use of different nesting levels of composite states within UML statechart diagram affect the understandability of the diagrams?" They refine the question to three null-hypotheses and use a previously defined metrics counting the nesting level in composite states (NLCS). They perform a controlled experiment with 38 subjects and 1 object (a statechart diagram of an ATM machine) described using three level of nesting (0, 1, and 2). The results of the preliminary controlled experiment show a trend towards flat statechart diagrams to ease program understanding.

However, as indicated by José and further highlighted by the workshop participants, the results presented are threaten by several different threats. In particular, the controlled experiments tested the understanding of the subjects (which are, really, the objects of the experiment due to their number) instead of the ease of understanding the ATM machine UML statechart diagram (which is, really, the subject of the experiment) because only one such diagram was used. Moreover, no independent measure of the complexity of the statechart diagram was performed wrt. polymorphism vs. traditional complexity or the number of children. Another participant pointed out the need to include the level of granularity, considering whole programs and more metrics and considering the programming languages for OO (Smalltalk, Java, C++). Finally, a last participant underlined the need to evaluate the quality of the subject programs.

On the Relationship between Cyclomatic Complexity and the Degree of Object Orientation

Grégory Seront, Miguel Lopez, Valérie Paulus, and Naji Habra (presenter), CETIC & University of Namur

Naji presented a study of the well-known McCabe's cyclomatic complexity in the context of object-oriented programming languages. He and his colleagues assume that complexity in object-oriented programming is hidden by polymorphism and overloading mechanisms. They design an experiment to assess the relations between the degree of object-orientation of programs (data abstraction, encapsulation, polymorphism, and inheritance) wrt. the cyclomatic complexity to test if the cyclomatic complexity is inversely correlated to the degree of object-orientation. On a sample of 694 programs, they relate the DIT as representative of the degree of object-orientation with the WMC as representation of the cyclomatic complexity. They find that no correlation exist between DIT and WMC for the 694 programs.

They conclude, however, that these results are preliminary and do not invalidate their assumption, as participant also pointed out. Indeed, DIT might now be representative enough, per se, of the degree of object-orientation. More experiments are necessary to conclude on the assumption.

The Advisability of using Packages in Data Warehouse Design

Manuel Serrano (presenter), Rafael Romero, Juan Carlos Trujillo, and Mario Piattini, Univ. Castilla La Mancha & Universidad de Alicante

Manuel presented an empirical study of the use of packages in UML class diagrams, when modeling data warehouses. The goal of the study is to determine whether or not the use of packages improve understandability. Using two groups of subjects (originally at University of Allicante, with a replication study at University of Castilla-La Mancha), and two objects (with and without packages) semantically equivalent, Manuel and his colleagues measure the effectiveness and the efficiency (as defined in their paper) of the two groups. They conclude, after analysis of the data, that there is no difference between models with and without packages wrt. understandability. However, the results do not, at this point, invalidate the assumption. Indeed, the authors as the participants highlighted that the number of subjects and of objects and the type of objects are possible threats to the validity of the study. Moreover, no hypothesis is made in relation with the layout of the diagrams, which has an impact on the understandability as well as with the decomposition of the models in packages (even though the data warehouse development methodology provides direction for the decomposition). Also, participants emphasized the need to focus more the study, in particular with respect to the role of the study supervisor (should they answer questions?, what questions?, how?). Other concerns focus on the use of pen and paper during the study (instead of software) and on the measure definitions. Indeed, the measure defini-

tions might need to be weighted (number of correct answer wrt. the number of question as in Antoniol at ICSM'05).
Finally, the assumption of the study seems just "obvious"; participants ask if it was interesting to test such an intuitively "obvious" hypothesis even though no formal proof has been given. The surprising results of the study seem to show that it is interesting to evaluate such "obvious" hypothesis.

Comparing the Results of Relation Analysis and Coupling Metrics -- Initial Case Study

Jonne Itkonnen, University of Jyväskylä

Jonne give a short presentation on the relations between Relation Analysis (RA) and coupling metrics. RA is a technique to identify parts of a program that contain unspecified logical couplings, couplings that exist only in the developers' minds. Coupling metrics measure specified couplings between parts of programs, i.e., couplings existing explicitly in the program declaration. An experimental study is performed to assess the correlation between RA and couplings metrics (represented by instability and abstractness measures) measured on one program, in which three classes seem to have unspecified logical constraints. The first result of the study is that some relations exist between RA and coupling metrics. However, more experiments should be performed to confirm the results, as stressed by the author and participants.

Towards a Multi-paradigm Complexity Measure

Zoltan Porkolab and Adam Sillye
For unexpected personal reasons, this position was not presented.

6 Discussions

The workshop finished with a discussion and closing session. The first part of this session was devoted to the identification of important issues that emerged from the previous sessions of the day.

The first issue is the need for quality models altogether. Participants stress that quality models are required to put some order in programs and, thus, that they are necessary. However, other participants emphasize the generality of quality models, such as IEEE or ISO models, which must be tailored for each purpose, tailored. General models must include everything (including human factors) so that everyone agrees. Moreover, general quality models are not applicable for ubiquitous UML, because UML is not a final product. However, some participants challenge the need for agreed-upon definitions, in particular because it is such a hard challenge. Right now, people outside of the research community have different interpretation of general quality models and (re)define new vocabulary, quality meta-models, frameworks to build quality models, but this represent the danger of having different concepts with same names or of aggregating unrelated metrics. Participants propose to counterbalance the generality of quality models, with industrial partners, through the use of tailor mechanisms depending on the context of use and using ontologies (thus, highlighting the need for repositories of concepts and relations) . Also, general quality models are usually hierarchical, while relation models are better (in particular wrt. thresholds and combinations).

The second issue relates to the use of structural metrics wrt. program semantics. Participants argue that metrics help in defining and in understanding the semantics of programs because metrics are a kind of semantics, even though only a summary, a first phase to detect something. Indeed, participants recall that if something smells, looks, and tastes like "A" then it is (most likely) "A". Metrics can be used to analyze the semantics through data mining techniques, i.e., rule-based semantics. However, it is unclear how to incorporate the context, set of data used for building models. Moreover, participants highlight problems when using metrics to study semantics: Generalization is difficult because of many hidden assumption and the lack of definition of the studied abstractions.

The third issue concerns the use of threshold values and the combination of metrics. Indeed, even if they are many ways to deal with thresholds (fuzzy logic, decision tree, CBR, model adaptation, visualization), general values are still use, in particular in the industry, even though thresholds do not apply in another contexts or systems. Participants question the use of thresholds and of metrics as indicators altogether, which is a paradox for a community interested in metrics. However, participants stress the need in the industry for values to decide at the end, because it seems so much more "certain". We face complicated problems; a single value is not enough. We must explain in the industry that it's not black-box: Lots of coupling or little coupling is only a trend, depending on the nature of the problem, because coupling can be a problem in a particular context only.

The last issue discussed among the participants was on the use of metrics (also with respect to semantics). For example, some participants attempt to use metrics with refactorings, but metrics do not always help in knowing the situation that you are looking for, because refactoring depends on the program. Thus, participants question the efficiency of metrics wrt. refactorings. However, refactorings are supposed to preserve behavior, but what is behavior? There is a difference between behavior and semantics, which should help in filtering out non-candidate refactorings. However, you must consider sequences of refactorings, but each step might generate errors (local vs. global optimum). Participants highlight the possibility to simulate, using refactorings, chains of refactorings. Other participants question the use of metrics for other use, such as biometrics, clone detection.

In conclusion, the discussions focused on the use of software metrics for semantic analyses, for refactorings, for prediction... Also, the participants are concerned with the still-widespread use of threshold values in the industry and academia and with the definition, construction, and use of quality models in context.

6 Participants' affiliation and contacts

The following tables include detailed information about respectively the organizers and the participants in this workshop. Names in boldface indicate the participants present in workshop).

Organizers

Name	Affiliation	Email address
Fernando Brito e Abreu	Universidade Nova de Lisboa, Portugal	fba@di.fct.unl.pt
Coral Calero	Universidad de Castilla-La Mancha, Spain	Coral.calero@uclm.es
Michele Lanza	University of Lugano	michele.lanza@unisi.ch
Geert Poels	Ghent University	geert.poels@ugent.be
Houari Sahraoui	Université de Montréal	sahraouh@iro.umontreal.ca

Other participants

Name	Affiliation	Email address