Proyecto DYNAMICA

II Jornadas de Trabajo DYNAMICA

(DYNamic and Aspect-Oriented Modeling for Integrated Component-based Architectures)

En Málaga, 11 de noviembre de 2004

Organización:

Grupo de Ingeniería del Software y Sistemas de Información

ISeI

Colaboradores:

Universidad Politécnica de Valencia

Departamento de Sistemas Informáticos y Computación

ACTAS

II Jornadas de Trabajo DYNAMICA

(DYNamic and Aspect-Oriented Modeling for Integrated Component-based Architectures)

En Málaga, 11 de noviembre de 2004

Editores:

Nour Ali
Artur Boronat
Patricio Letelier
Jenifer Pérez

Grupo de Ingeniería del Software
y Sistemas de Información

ISaI

Financiado por el CICYT (Centro de Investigación Científica Y Tecnológica), Proyecto DYNAMICA (DYNamic and Aspect-Oriented Modeling for Integrated Component-based Architectures)

Presidente de las jornadas

Isidro Ramos Salavert

Comité Organizador

Presidente: **Patricio Letelier Torres**

Vocales: **Nour Ali Irshaid**

Artur Boronat Moll

Jennifer Pérez Benedí

Comité Técnico

Isidro Ramos Salavert, *Universidad Politécnica de Valencia*

Bárbara Álvarez Torres, *Universidad Politécnica de Cartagena*

Coral Calero Muñoz, *Universidad de Castilla-La Mancha*

Francisco José Rodríguez Urbano, *Universidad Carlos III*

Joaquín Lasheras, *Universidad de Murcia*

Subproyectos y Grupos Participantes

PRISMA – Plataforma OASIS para Modelos Arquitectónicos
TIC2003-07804-C05-01
Grupo de Investigación "Ingeniería de Software y Sistemas de Información" (ISSI)
Departamento de Sistemas Informáticos y Computación (DSIC)
Universidad Politécnica de Valencia (UPV)

SUBPROYECTO: ANCLA - Arquitecturas DiNámiCas para Sistemas de TeLeoperAción
TIC2003-07804-C05-02
Grupo DSIE (División de Sistemas e Ingeniería Electrónica
Departamento de Tecnologías de la Información y Comunicaciones
Universidad Politécnica de Cartagena (UPCT)

SUBPROYECTO: CALIPO: CALIdad en POrtales
TIC2003-07804-C05-03
Grupo ALARCOS
Departamento de Informática
Universidad de Castilla-La Mancha (UCLM)

SUBPROYECTO: CARATE - Controlador de Arquitectura Reconfigurable para aplicaciones de Teleoperación
TIC 2003-07804-C05-04
Departamento de Ingeniería de Sistemas y Automática (ISA)
Universidad Carlos III (UC3M)

SUBPROYECTO: PRESSURE - PREcise Software modelS and reqUirements REuse
TIC 2003-07804-C05-05
Grupo de Investigación en Ingeniería del Software (GIS)
Departamento de Informática y Sistemas
Universidad de Murcia (UMU)

Presentación

El Proyecto DYNAMICA (**DYN**amic and **A**spect-Oriented **M**odeling for **I**ntegrated **C**omponent-based **A**rchitectures) es un proyecto coordinado de tres años de duración financiado por el Ministerio de Ciencia y Tecnología. DYNAMICA surge por los intereses comunes de cinco grupos de investigación españoles: el grupo de Ingeniería del Software y Sistemas de Información (ISSI) de la Universidad Politécnica de Valencia (UPV), el grupo de División de Sistemas e Ingeniería Electrónica (DSIE) de la Universidad Politécnica de Cartagena (UPCT), el grupo ALARCOS de la Universidad de Castilla-La Mancha (UCLM), el grupo del Departamento de Ingeniería de Sistemas y Automática (ISA) de la Universidad Carlos III (UC3M) y el Grupo de Investigación en Ingeniería del Software (GIS) de la Universidad de Murcia (UMU).

Las segundas jornadas DYNAMICA nos proveen un espacio para compartir los resultados de este primer año del proyecto. Así, el objetivo de estas jornadas es promover la discusión de soluciones para enfrentar la dinámica del software desde una visión arquitectónica en dominios con problemas reales como los sistemas de teleoperación, los sistemas hidráulicos y los portales Web.

Se han recibido 16 contribuciones desde los diferentes nodos, reflejando el espíritu e ilusión que tenemos invertido en este proyecto. Además, las relaciones inter-nodos han sido intensas, mostrando que somos conscientes que es el único modo de conseguir un avance hacia un objetivo común es mediante el trabajo colectivo. La amistad y sinergia entre todos nosotros permitirá un excelente trabajo y un gran resultado en este proyecto.

Esperamos que las discusiones sean intensas y las disfrutemos al máximo.

Los Editores.

Índice

Sección 1: Análisis, Formalización y Transformación de Modelos

Sección 2: Métricas y Calidad

Sección 3: Arquitecturas Software y Dominios de Aplicación

Sección 5: Ingeniería de Requisitos

Finding "early" indicators of the understandability and modifiability of OCL expressions with UML/OCL models

Luis Reynoso
National University of Comahue, Argentine
lreynoso@uncoma.edu.ar

Marcela Genero, Mario Piattini
Alarcos Research Group
University of Castilla La Mancha, Spain
{Marcela.Genero, Mario.Piattini}@uclm.es

Abstract

The use of UML for building a model is not enough to express many design decisions, due many essential aspects cannot be specified using only diagrammatic notations. Instead, using UML/OCL combined models the modeler can make decisions at a high level of abstraction. Numerous studies have pointed out that these early decisions and the quality of conceptual models have a high impact on the OO software system which is finally delivered. However, even though the use of UML/OCL combined models is essential for building higher quality models, there is no defined metrics for OCL expressions. For that reason in a previous work we have defined and theoretically validated a set of metrics for OCL expressions within UML/OCL models. The aim of this paper is to present a controlled experiment we have carried out in order to ascertain if any relation exists between two factors, length and coupling, measured by the Depth of Navigations (DN) and number of Navigated Classes (NNC) respectively, and two maintainability sub-characteristics: understandability and modifiability. Through experimentation we found that DN is highly correlated with the time the subjects spent on understanding and modifying OCL expressions.

1. Metrics for OCL expressions within UML/OCL models

It is widely recognized that the quality of UML models is highly dependent on decisions made early in its development and has a relevant repercussion in the software product that is finally implemented, so special attention must be paid to the quality of artifacts produced at the initial stages of the software systems life-cycle [4], [5], [11]. This fact is corroborated by the huge amount of metrics that have been proposed in the literature which measures quality aspects of UML models [1], [6], [9].

The metrics defined until now do not allow to capture those design decisions (made early during software development) that cannot be expressed using only graphical notations [10], [19]. However, with the introduction of OCL [15] by OMG, the quality of a UML model can be improved specifying it in a combination of the UML and OCL languages, i.e., through a UML/OCL combined model. A UML/OCL combined model is considered a complete, detailed, consistent and precise description of a system of a high level of maturity, its OCL expressions are written referencing to the model elements, constraining, queering and defining semantic properties of them. Without OCL expressions the model

would be severely underspecified [19]. This led us to think about the necessity of having metrics to measure structural properties for OCL expressions of UML/OCL combined models.

The lack of metrics for OCL expressions motivated us to define as set of metrics for measuring the structural properties of OCL expressions which are specified within UML class diagrams [16], with the idea that could be useful indicators of the understandability and modifiability of OCL expressions. We hypothesized that OCL expression structural properties have an impact on the cognitive complexity of modelers, due to the fact that when developers try to understand an OCL expression (considered in our study as a single mental abstraction: a chunk) they apply cognitive techniques, such as "chunking" and "tracing" [8]. "Tracing" has been observed as a fundamental activity in software comprehension [2], [14]. For that reason, we have defined metrics related to these cognitive techniques in [16]. A high cognitive complexity leads to an OCL expression reducing its understandability, and this conduce to undesirable external qualities, such as reduced maintainability and increasing fault-proneness. The metrics were defined following a methodology [7] which consists of three main step: metric definition, theoretical and empirical validation.

The main purpose of this paper is to describe a controlled experiment carried out to corroborate if any relation exists between two factors, length and coupling, measured by the Depth of Navigations (DN) and number of Navigated Classes (NNC) respectively, and two maintainability sub-characteristics: understandability and modifiability. Both metrics were validated as coupling interaction based and a length metric respectively in [16].

The definition of these metrics is provided in table 1. DN metric is related to the "tracing" technique due to its use of an important OCL concept: the navigation, whereas NNC is related to the "chunking" technique because it involves the comprehension of a class as a chunk. Although the DN metric was validated as a length metric its meaning is closely related to coupling concepts as it is shown in its definition (see table 1).

Coupling is one of the most complex software quality attribute in OO systems, and there are many different mechanisms that can constitute coupling [3]. A high quality software design, among many other principles, should obey the principle of low coupling [3]. Any navigation used in an OCL expression expresses coupling information between the contextual instance and other objects in a model. In fact, the metric DN reveals how dependent the contextual instance of distant objects in a class diagrams is, while NNC reveals how many different objects are coupled to the contextual instance.

Table1. Metric definition

Number of Navigated Classes (NNC)
Definition: This metric counts the total number of classes, association classes or interfaces to which an expression navigates to (these classes are frequently used in an expression through relationships role names). If a class contains a reflexive relation and an expression navigates it, the class will be considered only once in the metric. **Goal:** Warmer and Kleppe [19] argue that "any navigation that traverses the whole class model creates a coupling between the objects involved". A higher number of navigated classes will increase the coupling

between the objects.
Depth of Navigations (DN)
Definition: Given that in an OCL expression there can be many navigations regarding its definition, we build a tree of navigation using the class names used in navigations. We will only consider navigations starting from the contextual instance (from self). The root of the tree is the class name which 'self' represents. Then we build a branch for each combined navigation, where each class we navigate to is a node in the branch. Nodes are connected by "navigation relations". DN is defined as the maximum depth of the tree. **Goal:** A higher depth of navigations may involve a complicated navigation. Warmer et al. [19] suggest avoiding complex navigation expressions, and they also argue that: "using long navigation makes details of distant objects known to the object where we started the navigation". It is based on the idea that a higher value of this metrics will be an indicator of how distant the objects known by the Classifier where the expression is defined are.

In relation to our aim the following section describes how we carried out a controlled experiment. Finally, in section 3 some conclusions are drawn and future work is described.

2. A controlled experiment

The main goal of the experiment is to ascertain *if any relation exists between two factors, length and coupling, measured by DN and NNC respectively, for the purpose of evaluating the capability of using both metrics as understandability and modifiability indicators of OCL expressions.* In other words, if such relation exists we will have found, to some extent, early indicators of OCL expressions understandability and modifiability. We have followed some suggestions provided by Wohlin et al. [20] and Juristo and Moreno [12] on how to perform a controlled experiment.

Hereafter, we will briefly describe the main characteristics of the experiment:

- **Subjects.** In order to select the subjects we motivated a group of students who had taken a semester on System Analysis (at the National University of Comahue, UNC, Argentina) to take an additional and intensive course about OCL language. In the last lecture of the course where the experiment was run, fifteen students participated. The profile of the subject was the following: their average age is 24 years old, they have an average of 4 years of experience in programming, and one year in modeling with UML.
- **Variable selection.** The independent variables were two structural properties: length and coupling. The dependent variables were two maintainability sub-characteristics: understandability (UND) and modifiability (MOD) of OCL expressions.
- **Instrumentation.** The objects were four UML/OCL combined models, each of them having one OCL expression. These models were related to different universes of discourse that were easy enough to be understood by each of the subjects. Each model had a test enclosed that included two types of tasks:
  - UND tasks: They consist of four questions about the meaning of the OCL expression within the UML/OCL model. These questions reflected whether or not the subjects had understood each diagram. The subjects also had to write down the time they spent on answering the questions (UND time).
  - MOD tasks: Each subject had to modify the OCL expression according to three new requirements. The modifications to each test were similar, including defining new

navigations, attributes referred through navigations, etc. The subjects were required to write down the time they spent on performing the modifications (MOD time).

The independent variables were measured through the DN and the NNC metrics. The dependent variables were measured through the UND time and the MOD time, respectively.

- We also design a debriefing questionnaire including personal details and experience of the subjects which help us to identify their profile.

- **Hypotheses formulation.** We wish to test the following hypotheses:
  - Null hypothesis, $H_{0,1}$: There is no effect of length (measured by DN) on the UND time of OCL expressions. // $H_{3,1}$: $\neg H_{0,1}$
  - Null hypothesis, $H_{1,1}$: There is no effect of coupling (measured by NNC) on UND time of OCL expressions. // $H_{4,1}$: $\neg H_{1,1}$
  - Null hypothesis $H_{2,1}$: There is no interaction effect between length (measured by DN) and coupling (measured by NNC) on UND time of OCL expressions. // $H_{5,1}$: $\neg H_{2,1}$

  Analogously hypotheses for the MOD time are defined, named as $H_{i,2}$, i=0..5.
- **Experiment design.** Taking the hypotheses into account, we considered two factors: NNC and DN metrics, having each two values (low, high), 1 and 3 for DN, and 2 and 4 for NNC. The 2x2 crossed factorial design is shown in Table 2. We selected a within-subject design experiment, i.e., all the tests (experimental tasks) had to be solved by each of the subjects. The tests were put in a different order for each subject for alleviating learning effects.

Table 2. A 2X2 crossed factorial design

		DN	
		Low	High
NNC	Low	2,1 (Test 1)	2,3 (Test 3)
	High	4,1 (Test 2)	4,3 (Test 4)

- **Execution:** The experiment was run in one session. The subjects were given all the materials previously described. We explained to them how to carry out the tests, asking for carrying out the test alone, and using unlimited time to solve it. There was an instructor who supervised the experiment and any doubt could be asked to him.
- **Data validation.** In order to check if the collected data was correct the UND correctness (Number of correct answers /Number of questions answered) and MOD correctness (Number of correct modifications/Number of modifications applied) indicators were used to validate the data. We used box-plot in order to identify any possible outliers, representing unusual data points in the collected data due to an excessive time of subjects performing the UND or MOD tasks (usually they correspond to few subjects with learning or fatigue problems).

  After we collected the data we found that all data was complete. However, in the understandability part we decided to discard 20 tests (of 5 subjects) whose UND correctness was lower than 0.75 (only 2 out of 4 questions were correctly answered in

all the discarded tests). We think it was reasonable to discard these tests because the level of correctness is important in order to obtain reliable and relevant results about understandability aspects. Analyzing the data obtained, one outlier was identified. So, 36 tests of the UND tasks were analyzed. Regarding to the MOD tasks, the tests belonging to one subject was discarded due their correctness was lower than 0.75. In this case, three outliers were identified. So, 44 tests were analyzed.

- **Analysis and Interpretation**[1]. Before testing the formulated hypothesis we evaluated if the data follow a normal distribution or not using the Shapiro–Wilk test. We decided to select α =0.05 which means a 95% level of confidence. Due the data for the experiment was normal, we decided to carry out an ANOVA with repeated measures [18], because this type of analysis allows us to analyze interaction between the independent variables under study and the measurement of the dependent variable is repeated.
  Through the ANOVA we obtained that the DN metric has a great influence on the UND time and MOD time of OCL expressions. These findings reveal, to some extent, that OCL expressions that have in their definition "longer" navigations take more time to understand and more time to modify than others having "shorter" ones.

3. Conclusions and Future Work

Performing empirical validation with any defined metric is fundamental in order to demonstrate its practical utility. As many authors mentioned [13], [17] empirical validation employing experiments or case studies is fundamental to assure that the metrics are really significant and useful in practice. For that reason, in this paper, we have presented a controlled experiment for assessing if two metrics we proposed in [16] are closely related with OCL expressions understandability and modifiability. These metrics are Depth of Navigation (DN) and the Number of Navigated Classes (NNC). Both metrics are defined in terms of a fundamental OCL concept related to coupling: the navigation.

From the experimentation process we can conclude that the length factor measured, by the depth of the navigation (DN metric), emerges as the most important indicator for both dependent variables, and could be an "early" indicator of OCL expressions understandability and modifiability. Nevertheless, these findings must be considered as preliminaries, more experiments would be necessary in order to obtain more conclusive results. As a future work, we will do a replication of the experiment with professionals in order to increase the external validity of the results, and we will work in a generalization of the benefits of the set of metrics defined for OCL expressions, trying to obtain a global complexity of a class complemented with a set of OCL expressions (note that all the proposed metrics are defined in term of a single expression), aggregating in a upper immediate level a general result represented by a set of OCL expressions.

[1] All the data analysis was carried out by means of SPSS [19].

Acknowledgements
This research is part of the MESSENGER project (PCC-03-003-1) financed by "Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha (Spain), the CALIPO project supported by "Dirección General de Investigación del Ministerio de Ciencia y Tecnologia (Spain)" (TIC2003-07804-C05-03), the network VII-J-RITOS2 financed by CYTED, and the UNComa 04/E048 project financed by "Subsecretaría de Investigación de la Universidad Nacional del Comahue". Luis Reynoso enjoys a postgraduate grant from the agreement between the Government of Neuquén Province (Argentina) and YPF-Repsol.

References

[1].J. Bansiya and C. G. Davis. A Hierarchical Model for Object-Oriented Design Quality Assessment, IEEE Transactions on Software Engineering, Vol. 28 N° 1, January, 2002, pp.4-17.

[2].D. A. Boehm-Davis, J. E. Fox, and B. Philips. Techniques for Exploring Program Comprehension. Empirical Studies of Programmers, Sixth Workshop. Eds. W. Gray and D. Boehm-Davis. Norwood, NJ: Ablex, 1996, pp. 3-37.

[3].L. C. Briand L., C. Bunse and J. W. Daly. A Controlled Experiment for evaluating Quality Guidelines on the Maintainability of Object-Oriented Designs. IEEE Transactions on Software Engineering, Vol. 27 N° 6, June 2001, pp. 513-530.

[4].L.C. Briand, J.W. Daly, and J. K. Wüst. A Unified Framework for Coupling Measurement in Object-Oriented Systems, IEEE Transactions on Software Engineering, Vol. 25 N° 1 January 1999, pp. 91-121.

[5].L. C. Briand, S. Morasca, and V. R. Basili. Defining and Validating Measures for Object-based High Level Design. IEEE Transactions on Software Engineering. Vol. 25 N° 5, 1999, September 1999, pp. 722-743.

[6].L. C. Briand. and J. Wüst. Modeling Development Effort in Object-Oriented Systems Using Design Properties. IEEE Transactions on Software Engineering, Vol. 27 N° 11, November 2001, pp. 963-986.

[7].C. Calero, M. Piattini, and M. Genero. Method for obtaining correct metrics. Proc. of the 3rd International Conference on Enterprise and Information Systems (ICEIS'2001), pp. 779-784. 2001.

[8].S. N. Cant, B. Henderson-Sellers, and D. R. Jeffery. Application of Cognitive Complexity Metrics to Object-Oriented Programs. Journal of Object-Oriented Programming, Vol. 7. N° 4, pp. 52-63. 1994.

[9].D. Card, K. El-Emam and B. Scalzo. Measurement of Object-Oriented Software Development Projects, Software Productivity Consortium NFP, January 2001.

[10]. S. Cook, A. Kleepe, R. Mitchell, B. Rumpe, J. Warmer and A. Wills. The Amsterdam Manifiesto on OCL. Tony Clark and Jos Warmer, editors, Advances in Object Modelling with the OCL, pp. 115-149. Springer, Berlin, LNCS 2263. 2001.

[11].M. Genero. Defining and Validating Metrics for Conceptual Models, PhD Thesis, University of Castilla-La Mancha. 2002.

[12].N. Juristo and A. Moreno. Basics of Software Engineering Experimentation. Kluwer Academic Publishers. 2001.

[13].B. Kitchenham, S. Pflegger, and N. Fenton. Towards a Framework for Software Measurement Validation. IEEE Transactions of Software Engineering, Vol. 21 N° 12, December 1995, pp. 929-944.

[14].T. Klemola. A cognitive model for complexity metrics. 4th International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering. Sophia Antipolis and Cannes, France. 2000.

[15]. Object Management Group. UML 2.0 OCL 2nd revised submission. OMG Document ad/2003-01-07. [On-line] Available: http://www.omg.org/cgi-bin/doc?ad/2003-01-07.

[16].L. Reynoso, M. Genero and M. Piattini, (2004). -Measuring OCL Expressions: An approach based on Cognitive Techniques. Piattini M., Genero M. and Calero C. (Eds.), Imperial College Press, UK (to appear).

[17].N. F. Schneidewind. Methodology For Validating Software Metrics. IEEE Transactions of Software Engineering, Vol. 18 N° 5, May 1992, pp. 410-422.

[18].SPSS, 2002 SPSS 11.5. "Syntax Reference Guide". Chicago. SPSS Inc. (2002)

[19].J. Warmer and A. Kleppe. The Object Constraint Language. Second Edition. Getting Your Models Ready for MDA. Object Technology Series. Addison-Wesley. Massachusetts. 2003.

[20].C. Wohlin, P. Runeson, M. Höst, M. Ohlson, B. Regnell, and A. Wesslén. Experimentation in Software Engineering: An Introduction, Kluwer Academic Publishers. 2000.