IMTC/06

Proceedings of the 23rd IEEE Instrumentation and Measurement Technology Conference

Sorrento - Italy
24-27 April 2006

IEEE Catalog Number:
06CH37714C
ISBN : 0-7803-9360-0

For technical inquiries:

Pasquale Daponte
Phone - +39 0824 30 58 17
Fax - +39 0824 30 58 40
E-mail - daponte@unisannio.it

IMTC/06®

Proceedings of the

IEEE
Instrumentation and Measurement Technology Conference

Hilton Sorrento Palace, Sorrento, Italy
24-27 April 2006

Organized and sponsored by the IEEE Instrumentation and Measurement Society

IEEE Catalog Number 06CH37714C ▪ ISBN 0-7803-9360-0

Additional copies may be ordered from:

IEEE Service Center
445 Hoes Lane
Piscataway, NJ 08855-1331 USA

IEEE Catalog No. 06CH37714C
ISBN: 0-7803-9360-0

Printed in the U.S.A.

©2006 IEEE

Welcome to A View on the New Technologies for Instrumentation and Measurements

On behalf of the organizing committee, we cordially welcome you to the 2006 IEEE Instrumentation and Measurement Technology Conference (IMTC).

This IMTC edition will be held for the first time South of Italy - Sorrento. The IMTC venue, Hilton hotel, with its modern yet inviting facilities, stands on a hill overlooking the town of Sorrento and offering to its guests a unique and spectacular view, beautifully framed by the jagged outline of the Gulf of Naples on the horizon, the unmistakable silhouette of Mount Vesuvius, while just round the corner unfolds the breathtaking Amalfi Coast. The same encounter of sky, land and sea that has inspired so many poets and writers since the old Roman times.

IMTC/2006 continues a series of successful scientific and technical annual events with this 23rd edition. The conference covers all aspects of the theory and practice of instrumentation, measurement and control technologies, and related applications. This year the conference is focused on *A View on the New Technologies for Instrumentation and Measurements*. The impetuous escalation of the technology requires, as a consequence, a continuous view on its impact on instrumentation and measurement field both from research, industry, and daily life.

We are sure that the universally recognized beauty of Sorrento will be a perfect frame for this prestigious conference. It is, in fact, a further occasion, not only to meet old friends and new people from all over the world, but, moreover, to engage with them a continuous comparison directed to make wider the views on the technological progress of Instrumentation and Measurement.

The IMTC/2006 organization was a complex task due to the large and increasing interest of our research and application areas. Effort from many people was required to shape the technical program, arrange accommodation, manage the administrative aspects, and set up the social functions. We like to take this opportunity to thank all and each of them. We like also to thank the public and private organizations that supported the meeting in different ways.

For the first time a complete automatic web based abstract selection was introduced, the final electronic version of the manuscripts was formatted by using an adequate software developed for this aim.

The IMTC/2005 Technical Program consists of 86 oral and poster sessions scheduled over three days. With the wide range of technical sessions covering the whole range of the Instrumentation and Measurement field we are happy to welcome you to the variety of technical presentations that await you this year. Thanks to all of the Technical Program Committee members and the reviewers who have contributed to make this outstanding program possible.

The technical program was particularly difficult to be arranged since we received, for the first time 644 abstracts from all over the world. Due to the time limits of the conference only 460 papers have been selected after a painstaking activity of the program committee and additional reviewers. We like to thank all people who contributed to this process with opinions, comments, and suggestions to choose the best papers and even improve their quality.

The technical program encompasses several events and activities. The keynote speech will be held by FERRARI experts in the field of electrical and electronic measurements, we tried to show a one the well recognized world wide Italian technology.

Several parallel sessions will accommodate the contributed papers: to avoid overlapping of sessions on related topics, papers are distributed in separate thematic tracks. A special one, the 2nd ADC Forum, was devoted to ADCs, DACs and DAQ. Some sessions are designated as special since they have been solicited and organized and will be chaired by well-known experts from the international industrial and academic communities. Panels have also been organized to allow a wider and in-depth discussion of some hot topics related to education and measurement system modernization.

Tutorial sessions have been included to offer attendees hands-on, practical information.

Several Awards offered by International Institution and Companies will be assigned, in particular to young researchers.

The Conference is about to begin. You are now in a position to enjoy the fellowship of colleagues and experts and to pass free time in natural and artistic beauties. It is up to you to appreciate the Conference worth! Be critical! We, metrologists, colleagues, and friends, we know that this is the best way to improve quality, and to achieve lasting excellences.

General Chairman
Pasquale DAPONTE, University of Sannio,
Benevento, Italy

General Co- *Chairman*
Tom LINNENBRINK, IEEE TC-10 chairman,
Hittite Microwave Corp. Business Development Manager

Technical Program Co-Chairman
Fabrizio RUSSO, University of Trieste, Italy

Technical Program Co-Chairman
Kim FOWLER, IEEE I&M Magazine, Editor-in-Chief

TABLE OF CONTENTS

IMTC 2006 – Instrumentation and Measurement
Technology Conference
Sorrento, Italy 24-27 April 2006

Measuring Object Coupling in OCL Expressions: A Cognitive Theory-Based Approach

Luis Reynoso[1], Marcela Genero[2], Esperanza Manso[3]
[1]Department of Computer Science, Universidad Nacional del Comahue
Buenos Aires 1400, 8300 Neuquén, Argentina
Phone: +54-299-4490312, Fax: +54-299-4490313, Email: l_reynoso@uncoma.edu.ar
[2]ALARCOS Research Group, Department of Computer Science, University of Castilla La Mancha
Paseo de la Universidad 4, 13071 Ciudad Real, Spain
Phone: +34 926 295 300 ext 3740, Fax: +34 926 295 354, Email: Marcela.Genero@uclm.es
[3]GIRO Research Group, Department of Computer Science, University of Valladolid
Campus Miguel Delibes, E.T.I.C., 47011 Valladolid, Spain
Phone: +34 983 423 710 ext 25617, Fax: +34 983 423 671, Email: manso@infor.uva.es

Abstract – *Coupling is considered one of the most complex software quality attributes in Object Oriented (OO) Systems. We belief that OCL expressions provide essential coupling information at early phases of software development. For that reason, we started to study if object coupling information obtained from OCL expressions affects the quality of the model that is finally delivered, based on the hypothesis that object coupling (an structural property) of an OCL expression within a UML/OCL model (artifacts) can influence on the cognitive complexity of modelers (subjects), and high cognitive complexity leads to OCL expressions exhibit undesirable external qualities, such as reduced comprehensibility and modifiability. In this paper we explain the influence of object coupling on the comprehensibility and modifiability of OCL expressions following a cognitive theory-based approach. We study cognitive complexity using mental and cognitive models to explain the rationale of the measures we defined for measuring object coupling. This cognitive theory-based approach was also used for obtaining a clear explanation of a family of experiments we run to confirm our hypothesis. The empirical results reveal that there is empirical evidence that object coupling is strongly correlated with the comprehensibility and modifiability of OCL expressions.*

Keywords – *software measures, cognitive theory, software comprehension, coupling measures, OCL expressions, comprehensibility, modifiability, maintainability.*

I. INTRODUCTION

Within the more relevant theoretical basis for developing quantitative models relating products measures and external quality attributes [1], there is a strong hypothesis that an effect of measures on external quality attributes exists. In the software engineering literature, the mechanism causing this effect is assumed to be the cognitive complexity [5]. Cognitive complexity is defined as the mental burden of the individuals (modelers, developers, testers, etc.) who have to deal with software artefacts. This effect is hypothesized as follows: structural properties of a software artifact (such as coupling, cohesion, size, etc.) have an impact on its cognitive complexity. High cognitive complexity leads to an artifact exhibiting undesirable external qualities (such as reduced maintainability).

Nevertheless, rarely in the empirical software engineering literature neither defined measures nor empirical works are explained using cognitive theory. That is, there no further explanation of how a defined measure is related to the mental model of an individual, or giving an instrumentation to quantify the cognitive load of individual when they deal with experimentation objects.

The purpose of this article is twofold:

- First, we present an explanation of how measures for object coupling in OCL expressions are related to both, dimension of comprehension [3] and cognitive complexity models [4].
- Secondly, we describe a family of experiments to show that object coupling can influence on the comprehensibility and modifiability of OCL expressions, through the explanation of the effect of the cognitive complexity.

The first goal is developed in section 2 whilst the second is developed in section 3 and 4. Finally, in section 5 we present the conclusions and outline the future work.

II. COGNITIVE ASPECTS OF MEASURES

Regarding the first purpose of the paper, we defined in [18] a set of measures for structural properties of OCL expressions, taking into account that OCL expressions are the principal building block of modeling with OCL, and the relevance of the OCL language [13] within Model-Driven Engineering [12]. Within MDE (such as Model-Driven Development and Model-Driven Architecture) models are the primary focus for software development, and platform-independent model (PIM) forms the foundation for all later design work, as a major determinant of the quality of the overall OO. Several authors [22] argue that only the combination of OCL and Unified Modeling Language [14] avoid obtaining under-specified models, due to OCL is used as a textual add-on of the limited expressiveness of diagram-based UML notation.

In order to explain the cognitive complexity of individuals when they deal with OCL expressions, we based our reasoning of the comprehension of OCL expressions in two main theories: mental models and cognitive models. We are conscious that the application of the former theory is not straightforward, due they were defined for program comprehension, and the OCL expressions are declarative without getting embroiled in implementations details [10] as a program does. However, we consider that the dimension are generally enough to be applied to the comprehension of a declarative language like OCL. In fact, the essence of program comprehension is identifying artifacts, discovering relationships, and generating abstractions [21]. We believe that, whatever the language be, imperative or declarative, the comprehension of its specifications (or products) involve different cognitive dimensions. However, we believe that mental models are not the unique element impacting in the cognitive complexity of modelers, also cognitive processes and temporary information structures in the subject's head are also used to from the mental model [20] and that is the reason we evaluate cognitive models. Both applied theories, mental and cognitive models are explained in the subsections A and B, whereas subsection C describes the main elements of the cognitive dimensions and the cognitive process that a modeler employed during the comprehension of OCL expressions.

A. Mental Models

A mental model, also referred to as schema, is a predictive representation of real world systems. Mental representations are created to reason about, explain, and predict the objects and information in the world, behaviour of systems, etc [17]. So, mental models play an important role in software comprehension and correspondingly in comprehension-related tasks, such as modification and debugging. Many studies in software comprehension were performed largely with procedural applications [6], [7]. However, recently a number of studies have tested elements of the above mental model structure on OO software [2].

Two important dimensions of program comprehension were used as a basis during the last years:

- Dimension of Scope of Comprehension: The breadth of familiarity with the program gained by the programmer during comprehension activities determine this dimension [7]. Littman et al. [11] found two strategies used by programmers concerning scope of comprehension, systematic and as-needed. Using the systematic strategy, the programmer attempts to gain a broad understanding of the program before carrying out modifications. On the other hand, using the as-needed strategy, the programmer does not attempt to understand the overall design of the program but concentrates instead on the functioning of selected local parts of the code that are critically involved in the modification. Erdos et al. [9] propose partial comprehension as the only feasible approach when systems are large or when deadlines have to be met.
- Dimension of Direction of Comprehension: This direction of comprehension activities refers to the cognitive models used to group approaches of how programmers comprehend software. Existing program understanding models agree that comprehension proceeds either bottom-up, top-down or a combination of the two (opportunistic) [20], [15].

Recently a number of studies have tested elements of the mental model structure on OO software. In particular, the more relevant mental model of a bottom-up approach is the Burkhardt et al.'s model [2] based on the Pennington's model [16]. In these models the foundation is the existence of these two program abstractions during comprehension, the program model and the situation model, being the program model formatted before the situation model, in a bottom-up manner. This program model contains text structure knowledge, where programmers first develop a control-flow abstraction of the program, sequence of operations, etc. Once the program model has been fully assimilated, the situation model is developed, and this requires knowledge of the application domain. The situation model encompasses knowledge about the following structural components:

1. Problem Objects: These objects directly model objects of the problem domain.
2. Relationships between Problem Objects: These consist of the inheritance and composition relation-ships between objects.
3. Reified Objects: An example of a computing, or reified, object is a string class, which is not a problem domain object per se. Reified objects are represented at the situation model level in as much as they are necessary to complete the representation of the relationships between problem objects, i.e., they bundle together program-level elements needed by the domain objects.
4. Main Goals: The main goals of the problem correspond to functions accomplished by the program viewed at a high level of granularity.
5. Client– server Relationships: Communication between objects corresponds to client – server relationships in which one object processes and supplies data needed by another object.
6. Data Flow Relationships: Communication between variables corresponds to data flow relationships connecting units of local plans within a routine.

B. Cognitive Models

Two cognitive process or techniques that are concurrently and synergistically applied in problem solving and were defined by Cant et al. in the Cognitive Complexity Model (CCM model) [4]. The CCM gives a general cognitive theory of software complexity that elaborates on the impact of structure on comprehension [8]. Although the study of Cant et al. has been considered a reasonable point of departure for understanding the impact of structural properties "on comprehensibility of code and the coding process", we believe that this model can also be applied to UML developers when they try to understand OCL expressions. The two cognitive processes are chunking and tracing.

Chunking involves the recognition of a set of declarations and the extraction of information from them, which is remembered as a chunk (a single mental abstraction), whereas tracing involves scanning, either forward or backwards, in order to identify pertinent chunks. The comprehension of a particular "chunk" is the sum of three components: (1) the difficulty of understanding the "chunk" itself; (2) the difficulty of understanding all the dependencies on the "chunks" upon which a particular "chunk" depends, and (3) the difficulty of "tracing" these dependencies to those "chunks" . When a method calls for another method to be used in a different class, or when an inherited property needs to be understood, are typical examples of where "tracing" is applied.

C. OCL expression' comprehension

We believe that different issues influence the comprehension of OCL expressions:

- Direction of Comprehension: We believe that during early phases of OCL expressions comprehension, modellers tended to utilize a bottom up direction, by reading the textual representation of the OCL expressions and then mentally grouping these statements into higher level abstractions. During the grouping or chunking the expression the modellers switched to the graphical UML diagram to which the expression is associated with, looking for artefacts referred in the OCL expressions, i.e. rolename of relationships, methods, attributes, etc. However, in the lasts phases of OCL expressions comprehension process the modelers increasingly used an opportunistic direction of comprehension. Nevertheless, the direction is primarily bottom up, the OCL expression trace the process of understanding through the model. In Table 1 we analyzed the bottom-up direction aspect and we present in the third and fourth columns a mapping of each measure with the component of the situation (or program) model to which applies in the Burkhardt et al.'s model. This mapping was obtained according to:
  - Measures related to navigations (NNR, WNN, DN) are considered as mechanisms of relationship between problem objects, e.g. through navigations it is possible to access objects related to the contextual instance.
  - Problem objects, i.e. objects of the problem domain, are identified through the number of navigated classes (NNC), the number of attributes or operations of the contextual type (NAS or NOS), and the number of user-defined data type attributes or operations (NUDTA or NUDTO), the number of contextual instance (implicitly –NIS- or explicitly -NES- specified), the number of properties postfixed by @pre, and the collection operation iterator variables (when they represent a collection of problem objects).
  - Collection operations as an important OO mechanism to manipulate object collections are considered a reified object, i.e. not problem domain object per se. Due to definition expressions are used to factorize a common subexpression are considered reified objects as well.
  - NAN, WNON and WNM measures are considered client-server relationships because the contextual instance has a reference to a property (attribute or operation) of other objects.
  - The number of OCL keywords, the number of let' variables, and the number of boolean and comparison operators are considered elementary operations of the Program Model.
  - If expressions are considered a control flow mechanism of the Program Model.
- Scope of Comprehension: We believe that modellers uses a partial or systematic strategy, during OCL expression comprehension i.e., she or he attempt to understand the overall design to which the OCL expression is related. However we think that OCL expression modifications demands a narrow scope of comprehension, and modellers should apply a more as-needed strategy of the surrounding classes of the contextual type.
- Cognitive Model and cognitive techniques: An OCL expression is a suitable "chunk" unit, which modellers should understand as a whole declaration constraining an aspect of the system being modelled. To understand an OCL expression, as a "chunk", UML developers must carry out "tracing" to understand some properties (attributes, operations or rolenames) belonging to other classes, for example when a operation is referred through messaging, navigation of relationships, etc. "Tracing" is a technique commonly applied by UML developers during the comprehension of OCL specifications. In general, the understanding of an OCL expression as a "chunk" involves a strong intertwining of "tracing" and "chunking" techniques.
  An example of "chunking": The contextual instance, i.e. self, provides a point of reference for the interpretation of an OCL expression. Whenever we use the self keyword, we are doing "chunking" of the expression.
  An example of "tracing": If an OCL expression includes a navigation for accessing an attribute (or operation) belonging to a class, the proper evaluation of the navigation will disrupt the understanding of the OCL expression. The evaluation of the navigation, will involve the understanding of the rolenames (or class names if rolenames are missing in the diagram) mentioned in the navigation, to look for them in the class diagram, to look for the attribute (or operations) mentioned through the navigation, and eventually to understand the OCL expressions associated to this attribute (or operation), if they have OCL expressions attached. Table 1, fifth column, summarizes the measures we defined [18] according to the technique they are related to.

Table 1: Mapping between Measures for OCL expressions and Burkhardt' Mental Model and Cant' CCM

Measure	Measure Description	Mental Model of Burkhardt		Cognitive technique of CMM
		Model	Relation	
NNR	Number of Navigated Relationships	Situation	Rel. between Problem objects	Tracing
NAN	Number of Attributes referred through Navigations	Situation	Client-server objects	Tracing
WNON	Weighted Number of referred Operations through Navigations	Situation	Client-server objects	Tracing
NNC	Number of Navigated Classes	Situation	Problem objects	Tracing
WNM	Weighted Number of Messages	Situation	Client-server objects	Tracing
NUDTA	Number of User-Defined Data Type Attributes	Situation	Problem objects	Tracing
NUDTO	Number of User-Defined Data Type Operations	Situation	Problem objects	Tracing
WNN	Weighted Number of Navigations	Situation	Rel. between Problem objects	Tracing
DN	Depth of Navigations	Situation	Rel. between Problem objects	Tracing
WNCO	Weighted Number of Collection Operations	Situation	Reified objects	Tracing
NEI, NII	Number of Explicit/Implicit Iterator variables	Situation	Problem objects	Tracing
NAS	Number of Attributes belonging to the classifier that Self represents	Situation	Problem objects	Chunking
NOS	Number of Operations belonging to the classifier that Self represents	Situation	Problem objects	Chunking
NVD	Number of Variables defined through <<Definition>> expressions	Situation	Reified objects	Chunking
N@P	Number of properties postfixed by @pre	Situation	Problem objects	Chunking
NES	Number of Explicit *Self*	Situation	Problem objects	Chunking
NIS	Number of Implicit *Self*	Situation	Problem objects	Chunking
NKW	Number of OCL KeyWords	Program	Elementary operation	Chunking
NVL	Number of Variables defined by *Let* expressions	Program	Elementary operation	Chunking
NIE	Number of *If* Expressions	Program	Control Flow	Chunking
NBO	Number of Boolean Operators	Program	Elementary operation	Chunking
NCO	Number of Comparison Operators	Program	Elementary operation	Chunking

III. A FAMILY OF EXPERIMENTS

Regarding the second purpose of this paper, we use a family of experiments to explain the influence of object coupling (measured by the measures we defined) on the maintainability of OCL expressions, through the cognitive complexity. Card et al. argue [5] that one way to operationalize cognitive complexity is to equate it with the ease of comprehending an object-oriented artifact. So, in order to estimate the cognitive complexity of OCL expression during experimentation, we gather the subject perception of the complexity of OCL expressions comprehension and modification, using linguistic labels. This instrumentation of the cognitive complexity was used to triangulate hypothesis between structural properties, cognitive complexity and external quality attributes, following the rationale of the theoretical basis for developing quantitative models of Briand et al. [1].

The experiments were run in May and June (2005) at the Technical University of Valencia (UPV) and Complutense University of Madrid (UCM) respectively. In order to study the influence of object coupling on the comprehensibility (COM) and modifiability (MOD) of OCL expressions we have considered not only the time subjects spent on the COM and MOD tasks required in the experimental material, but also their efficiency and their subjective perception of the difficulty when carrying out these tasks. We think that, both quantitative (COM and MOD efficiency) and qualitative (subject's rating) information is important to obtain more solid findings.

Experiment Hypotheses: We formulated different hypotheses based on our beliefs:

- Hypotheses 1: $H_{0,1}$ The ranks of the (COM or MOD) Eff do not differ from their expected value, i.e. the mean efficiency is the same for all the models. $H_{1,1} = \neg H_{0,1}$
- Hypotheses 2: $H_{0,2}$ There is no significant correlation between the measures defined for OCL expressions, related to object coupling and their (COM or MOD) Eff. $H_{1,2} = \neg H_{0,2}$
- Hypotheses 3: $H_{0,3}$ There is no significant correlation between the OCL expression measures related to object coupling and the (COM or MOD) SubComp. $H_{1,3} = \neg H_{0,3}$
- Hypotheses 4: $H_{0,4}$ The COM or MOD SubComp are not correlated with the COM and MOD Time. $H_{1,4}$: $\neg H_{0,4}$
- Hypotheses 5: $H_{0,5}$ The (COM or MOD) SubComp is not correlated with the COM and MOD Eff. $H_{1,5} = \neg$H0,5.

From the descriptive and exploratory analyses, we can conclude that the efficiency of subjects when they had to comprehend and modify the OCL expressions, increased as time passed.

Due to the size of the population from the UCM experiment the formulated hypotheses of the experiments could not be tested in the same way as we analyzed the hypotheses for the UPV experiment. However, we used the UCM data to confirm the findings obtained in UPV, gathering the whole set of data from both experiments (after using statistical tests needed to realize if this is possible and enable us to collect all the data). We found different results for the hypotheses that were confirmed once we gather the whole data set. We summarize the findings for each hypothesis in Table 1.

Table I. Synopsis of Hypotheses and the statistical test applied.

Relation between	Efficiency COMEff, MOD Eff	Time COM and MOD Time	Subjective Complexity COM and MOD SubComp
OCL Expression Measures	Hypotheses 2 Test: Spearman (1) the NNR, NNC, WNN, DN, WNCO and NEI measures have are significant correlated with the COM Efficiency (2) the NNR, WNN, DN and NCO have significant correlated with the MOD Efficiency		Hypotheses 3 Test: Spearman The NNR, NNC, WNN, WNCO, NEI and DN measures are significant correlated with the COM subjective complexity; and NNR, WNN, DN, NCO and WNCO measures are significant correlated with the MOD subjective complexity
COM Sub-Comp MOD Sub-Comp	Hypotheses 5 Test: τ Kendall The subjective ratings are influenced by the COM and MOD efficiency.	Hypotheses 4 Test: τ Kendall The subjects' subjective ratings (COM or MOD rating) are influenced by the time they spent comprehending or modifying the OCL Expressions, i.e. both times seem to affect their perception of the level of complexity of an OCL expression.	

The results reveal again that there is empirical evidence that object coupling defined in an OCL expression is strongly correlated with the maintainability of OCL expression (hypothesis 2), but these findings are strengthen by the following facts:

- Considering the COM and MOD tasks, the object coupling of OCL expressions impact in the cognitive complexity in different way (hypothesis 3), and this effect is explained as follows using the cognitive theory.
  - Considering the constituent of the mental model of Burkhardt, Problem objects (NNC, NEI), Relation of problems objects (NNR, WNN, DN) and Reified objects (WNCO) affects the COM tasks. However only Relation of problems objects (NNR, WNN, DN) affects MOD tasks. We think that in a systematic strategy during comprehension a broad familiarity of the expression and its contextual information should be gathered by the subjects, however during modification, in as needed strategy, the focus of the comprehension was only on relationship between problem objects.
  - Measures related to chunking and tracing affect UND tasks, and mainly tracing affect the MOD tasks.
- COM and MOD Subcomp is correlated to the COM and MOD Time (hypothesis 5) and also with COM and MOD Efficiency (hypothesis 5).
- The set of measures that are correlated with the maintainability of OCL expression (hypothesis 2) is almost the same as the set of measures correlated with COM and MOD sub-comp (hypothesis 3).

IV. CONCLUSIONS

Due to the quantitative models relating product measures and external quality attributes based their hypothesis in the impact that a software artefacts (such as its coupling) have an impact on its cognitive complexity and the effect that a high cognitive complexity leads to an artefact exhibiting undesirable external qualities (such as reduced comprehensibility and modifiability), we consider the cognitive complexity as a central aspect that should be analyzed during the definition of measures and their empirical validation. We evaluate cognitive complexity of modellers when they deal with OCL expressions within UML/OCL models taking into account two cognitive theories: mental model and cognitive process. In relation to the former theory we use two dimensions of comprehension (scope and direction) and we focus on bottom-up direction of comprehension of Burkhardt [3]. Considering the latter theory we explain two cognitive process of the cognitive complexity model of Cant et al.: chunking and tracing. These two cognitive theories were applied to achieve the twofold goal of this paper as follows:

(1) We had analyzed how the two cognitive theories are related to comprehension of OCL expression. We described the relationships between each defined measures and the constituent of the mental model of Burkhardt model, eg. NNC and NEI are related to "problem objects", NNR, WNN, DN are related to

"relation of problems objects" and WNCO is related to "reified objects", etc.

(2) Regarding its application in an empirical work, we described a family of experiments run in May and June (2005) at the Technical University of Valencia and Complutense University of Madrid respectively. In order to study the comprehensibility and modifiability of OCL expressions we have considered not only the time subjects spent on the comprehensibility and modifiability tasks required in the experimental material, but also their efficiency and their subjective perception of the difficulty when carrying out these tasks. We think that, both quantitative and qualitative measures provide more information for obtaining obtain more solid and credible findings. The results reveal that, to some extent, there is empirical evidence that object coupling defined in an OCL expression is strongly correlated with the comprehensibility and modifiability of OCL expressions (hypothesis 2). Nevertheless, we are conscious that further replications are necessary, especially with subjects and materials taken from industrial settings.

The two purposes of this paper show what Rilling et al. argues in [19]: the understanding attributes of cognitive process can lead to new software measures that allow the prediction of human performance in software development for assessing and improving the maintainability of software artifacts.

ACKNOWLEDGMENTS

This research is part of the ENIGMAS project (PBI-05-058) financed by "Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha", the CALIPO project supported by "Dirección General de Investigación del Ministerio de Ciencia y Tecnología (Spain)" (TIC2003-07804-C05-03) and the COMPETISOFT project (506PI0287) financed by "CYTED (Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo)".

REFERENCES

[1] L. C. Briand, J. Wüst, S. Ikonomovski and H. Lounis. A Comprehensive Investigation of Quality Factors in Object-Oriented Designs: An Industrial Case Study. International Software Engineering. Research Network technical report ISERN-98-29, 1998.

[2] J. M. Burkhardt, F. Detienne, S. Wiedenbeck. Object-oriented program comprehension: Effect of expertise, task and phase. In Empirical Softw. Engg., Vol. 7, pp. 115-156, Hingham, MA, USA, 2002. Kluwer Academic Publishers.

[3] J. M. Burkhardt, F. Detienne, and S. Wiedenbeck. The effect of object-oriented programming expertise in several dimensions of comprehension strategies. In IWPC '98: Proc. of the 6th International. Workshop on Program Comprehension, page 82, Washington, DC, USA, 1998. IEEE Computer Society.

[4] S. N. Cant, B. Henderson-Sellers, D. R. Jeffery: Application of Cognitive Complexity Metrics to Object Oriented Programs. Journal of Object Oriented Programming, Vol. 7, No. 4, pp. 52-63, 1994.

[5] D. N. Card, K. El Emam. Measurement of Object-Oriented Software Development Projects. January, 2001.

[6] C. L. Corritore and S. Wiedenbeck. Mental representations of expert procedural and object-oriented programmers in a software maintenance task. Int. J. Hum.-Comput. Stud., Vol. 50, No.1, pp.61-83, 1999.

[7] C. L. Corritore, S. Wiedenbeck. Direction and scope of comprehension-related activities by procedural and object-oriented programmers: An empirical study. In IWPC '00: Proc. of the 8th International Workshop on Program Comprehension, Washington, DC, USA, IEEE Computer Society, page 139, 2000.

[8] K. El-Eman. Object-Oriented Metrics: A Review of Theory and Practice. National Research Council Canada. Institute for Information Technology, 2001.

[9] K. Erdos and H. M. Sneed. Partial comprehension of complex programs (enough to perform maintenance). In IWPC '98: Proc. of the 6th International Workshop on Program Comprehension, Washington, DC, USA, IEEE Computer Society, 1998, page 98.

[10] A. Hamie. Enhancing the Object Constraint Language for More Expressive Specifications. Software Engineering Conference APSEC 99. Proc.. Sixth Asia Pacific. pp. 376-383, 1999.

[11] D. C. Littman, J. Pinto, S. Letovsky, and E. Soloway. Mental models and software maintenance. *J. Syst. Softw.*, Vol. 7, No. 4, pp. 341-355, 1987.

[12] Object Management Group. MDA The OMG Model Driven Architecture. Available: http://www.omg.org./mda/, 2002.

[13] Object Management Group. UML 2.0 OCL 2nd revised submission. OMG Document. Available at http://www.omg.org

[14] Object Management Group. UML Specification Version 1.5, OMG Document. Available at http://www.omg.org

[15] V. Mayrhauser and A. M. Vans. Identification of dynamic comprehension processes during large scale maintenance. In IEEE Trans. Softw. Eng., Piscataway, NJ, USA. IEEE Press. Vol. 22, pp. 424-437, 1996.

[16] N. Pennington: Comprehension strategies in programming. In E. Soloway and S. Iyengar, Eds., Empirical Studies of Programmers: Second Worshop. Ablex, Norwood, NJ, pp. 100-113, 1987.

[17] V. Ramalingam, D. L., and S. Wiedenbeck. Self-efficacy and mental models in learning to program. In ITiCSE '04: Proc. of the 9th annual SIGCSE conference on Innovation and technology in computer science education, New York, NY, USA, ACM Press, pp 171-175, 2004.

[18] L. Reynoso, M. Genero, M. Piattini. Measuring OCL Expressions: An approach based on Cognitive Techniques. Imperial College Press, UK. 2005.

[19] J. Rilling and T. Klemola. Identifying Comprehension Bottlenecks Using Program Slicing and Cognitive Complexity Metrics. Proc. of the 11th IEEE International Workshop on Program Comprehension. IEEE Computer Society, pp. 115-119, 2003.

[20] M. A. D. Storey, F. D. Fracchia, H. A. Muller. Cognitive design elements to support the construction of a mental model during software exploration. The Journal of Systems and Software, Vol. 44, No.3, pp.171-185, 1999.

[21] S. Tilley, S. Huang. A qualitative assessment of the efficacy of UML diagrams as a form of graphical documentation in aiding program understanding. SIGDOC '03: Proc. of the 21st annual international conference on Documentation. ACM Press, pp. 184-191, 2003.

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