D. Patel, I. Choudhury, S. Patel
and S. de Cesare (Eds)

2000 OOIS

6th International Conference on Object Oriented Information Systems

Springer

OOIS 2000

6th International Conference on Object Oriented Information Systems 18 - 20 December 2000, London, UK

Proceedings

Edited by
Dilip Patel
South Bank University, London, UK
Islam Choudhury
London Guildhall University, London, UK
Shushma Patel
South Bank University, London, UK
Sergio de Cesare
Brunel University, London, UK

Presented by

LONDON GUILDHALL UNIVERSITY

Supported by

LogOn Technology Transfer

Springer

Dilip Patel
School of Computing, Information Systems and Mathematics,
South Bank University, 103 Borough Road, London, UK

Islam Choudhury
School of Computing, London Guildhall University, 100 Minories, London, UK

Shushma Patel
School of Computing, Information Systems and Mathematics,
South Bank University, 103 Borough Road, London, UK

Sergio de Cesare
Department of Information Systems and Computing, Brunel University,
Uxbridge, London, UK

ISBN 1-85233-420-7 Springer-Verlag London Berlin Heidelberg

British Library Cataloguing in Publication Data
A catalog record for this book is available from the British Library.

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress.

Typesetting: Camera ready by contributors
Printed and bound by the Athenæum press Ltd., Gateshead, Tyne & Wear
34/3830-543210 Printed on acid-free paper SPIN 10790885

Preface

New object-oriented technologies have been conceived and implemented over the past decade in order to manage complexity inherent in information systems development. Research has spanned from information systems modelling languages (UML and OML) to databases (ODMG), from programming languages (Java) to middleware technology (CORBA). A more widespread use of the Internet has led to the emergence and integration of various other technologies, such as XML and database connectivity tools, allowing businesses to access and exchange information over the Internet. The main theme of OOIS 2000 was "Object-Technology and New Business Opportunities" and focused on research conducted in the area of effective information systems development for the promotion of e-commerce. Papers were invited from academics and practitioners. The thirty-nine papers accepted for OOIS 2000 are included in these proceedings. It is nice to see this year that the shift from centralised to distributed systems and the widespread access and use of the Internet has allowed the advent of new opportunities for businesses to exploit, in the form of e-commerce.

The papers included in the proceedings embrace the above themes and have been presented in the following categories:

- Databases and Programming Issues
- Modelling and Design Issues
- Electronic Commerce
- XML and CORBA Issues
- UML and Modelling Issues
- Architectures, Patterns and Visualisation
- Measurements

During the preparation of this conference many people contributed towards the success of OOIS 2000. We would like to acknowledge the hard work of the programme committee and additional reviewers in reviewing the papers with a very short turn around time. We are indebted to the staff and research students from the Centre for Information and Organisation Studies at South Bank University who have contributed enormously in terms of time and effort. Our gratitude goes to the organising committee. Last but not least we would like to thank Rebecca Mowat at Springer-Verlag for her patience with us.

Contents

MODELLING AND DESIGN ISSUES II

XML AND CORBA ISSUES I

UML AND MODELLING ISSUES I

UML AND MODELLING ISSUES II

Defining Complexity Metrics for Object-Relational Databases

Coral Calero, Mario Piattini, Marcela Genero

Departamento de Informática
Universidad de Castilla-La Mancha
Ronda Calatrava, 5
13071 Ciudad Real (Spain)
email: {ccalero, mpiattin, mgenero}@inf-cr.uclm. Fax: +34 926295300

ABSTRACT . New Object-Relational Databases Management Systems (ORDBMSs) are replacing existing relational ones. In spite of the high expressiveness, application systems built upon ORDBMS are more complex and difficult to maintain due to the mixing of both paradigms, the relational and the object-oriented. All these reasons made necessary to dispose on metrics for measuring the complexity of this kind of databases and controlling its correct design. However, not always it is easy to propose correct metrics. This paper describes the method used by the authors for proposing metrics and the process of feedback needed for obtaining them in a correct way.

KEY WORDS: Object-relational databases, metrics, complexity

1. Introduction

We are witnessing important advances in database technology; a new ``generation" of DBMS (Database Management System) is coming out, among which object-relational ones (e.g. Oracle 8, Informix Dynamic Server, DB2) stand out. Object-relational databases will replace relational systems to become the next great wave of databases (Stonebraker and Brown, 1999). This kind of DBMSs supports a more complex data model having a stronger influence on the overall application maintenance effort. Taking into account that software maintenance is the most expensive stage in the software industry, it is very important to have maintainability metrics for this new kind of databases.

Maintainability is achieved by three factors: understandability, modifiability and testability, which in turn are influenced by complexity (Li and Chen, 1987). We must be conscious, however, that a general complexity measure is "the impossible holy grail" (Fenton, 1994). Henderson-Sellers (1996) distinguishes three types of complexity: computational, psychological and representational, and for psychological complexity three components are considered: problem complexity, human cognitive factors and product complexity. The last one is our focus.

In the next section we summarize the features of object-relational databases. In section 3, we describe the used method for proposing metrics. Metrics definition and refinement are presented in section 4. Finally, section 5 summarizes the paper and draws our conclusions.

2. Object-Relational databases

Object-relational databases combine traditional database characteristics (data model, recovery, security, concurrency, high-level language, etc.) with object-oriented principles (e.g. encapsulation, generalization, aggregation, polymorphism, ...). These products offer the possibility of defining classes or abstract data types, in addition to tables, primary and foreign keys and constraints[1], as relational databases.

Besides, generalization hierarchies can be defined between classes (super and subclasses) and between tables, subtables and supertables. Table attributes can be defined in a simple domain, e.g. CHAR(25), or in a user-defined class as complex number or image, In figure 1 an example based on the presented in (Cannan, 1999) is shown.

```
CREATE TABLE house(
idhouse INTEGER,
price INTEGER,
rooms INTEGER,
size DECIMAL (8,2),
location address,
desc text,
front_view bitmap,
document doc,
seller employee,
PRIMARY KEY(idhouse));

CREATE TABLE person(
id INTEGER,
myhouse INTEGER,
PRIMARY KEY (id),
FOREIGN KEY (myhouse) REFERENCES house);
```

```
CREATE TYPE address AS(
street CHAR(30),
city CHAR(20),
state CHAR(2),
zip INTEGER) NOT FINAL;

CREATE TYPE employee AS(
name CHAR(40),
base_salary DECIMAL(9,2),
bonus DECIMAL(9,2))
INSTANTIABLE NOT FINAL
METHOD salary() RETURNS
DECIMAL(9,2);

CREATE METHOD salary() FOR employee
BEGIN
...
END;
```

Figure 1. Example of table definition in SQL:1999

3. Used method

As we have said previously, our goal is to define metrics for controlling object-relational databases maintainability. But metrics definition must be done in a methodological way, it is necessary to follow a number of steps for ensure the reliability of the proposed metrics. Figure 2 presents the method we apply for the metrics proposal (Calero et al., 2001).

[1] In this first approximation constraints are not considered for measure purposes.

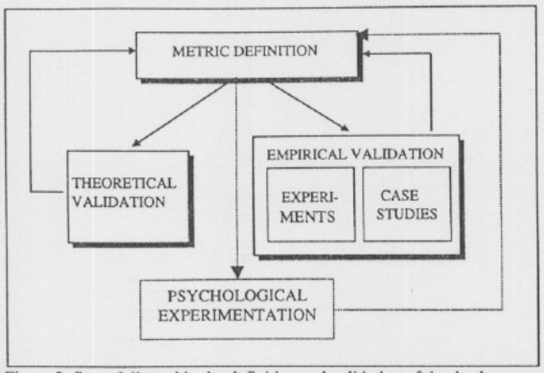

Figure 2. Steps followed in the definition and validation of the database metrics

In this figure we have four main activities:

- **Metrics definition.** The first step is the proposal of metrics. Although this step could look simple, it is necessary to take care on defining the metrics. This definition must be made taking into account the specific characteristics of the database we want to measure and the experience of database designers and administrators with these databases.
- **Theoretical validation.** The second step is the formal validation of the metrics. The formal validation help us to know when and how applying the metrics. There are two main tendencies in metrics validation: the frameworks based on axiomatic approaches and the ones based on the measurement theory. The goal of the first ones is merely definitional by defining formally desirable properties for measures for a given software attribute, so axioms must be used as guidelines for the definition of a measure. The most well-known frameworks of this type are those proposed by Weyuker (1988), Briand et al. (1996) and Morasca and Briand (1997). Software metrics axioms sets have been developed without a consensus and sometimes without a common understanding of the data to which they apply. The main goal of axiomatisation in software metrics research is the clarification of concepts to ensure that new metrics are in some sense valid. However, if an axiom set cannot itself be shown to be fit for purpose, it cannot be used to validate metrics. We cannot tell whether a measure that does not satisfy the axioms has failed because it is not a measure of the class defined by the set of axioms (e.g. complexity, length...) or because the axiom set is inappropriate. Since the goal of axiomatisation in software metrics research is primarily definitional, with the aim of providing a standard against which to validate software metrics, it is not so obvious that the risks outweigh the benefits (Kitchenham y Stell, 1997). The measurement theory-based frameworks (such as Zuse, 1998 or Withmire, 1998) specify a general framework in which measures should be defined. The strength of measurement theory is the formulation of empirical conditions from which we can derive hypothesis of reality.
- **Empirical validation.** The goal of this step is to prove the practical utility of the proposed metrics. There are a lot of ways to made it but basically we can divide the empirical validation in two: experimentation and case studies. The experimentation is usually made using controlled experiments and the case studies usually work with real data. Both of them are necessary, the controlled experiments for having a first approach and the case studies for making the results stronger. In both

cases, the results are analyzed using either statistics tests or advanced techniques as C4.5 a machine learning algorithm, and RoC a robust Bayesian classifier and so on. Also is necessary the experiment replication because the isolate results of an experiment it is difficult to understand how widely applicable the results are and, thus, to assess the true contribution to the field (Basili, 1999).

- **Psychological explanation**. Ideally we will be able to explain also the influence of the values of the metrics from a psychological point of view. Some authors, such as Siau (1999), propose the use of cognitive psychology as a reference discipline in the engineering of methods and the studying of information modeling. Cant et al. (1995) and Klemola(2000) propose the use of cognitive models to define object-oriented complexity metrics. In this sense, cognitive psychology theories (such as the Adaptive Control of Thought (ACT), Anderson (1983)) could justified the influence of certain metrics in the database understandability. The knowledge of the limitation of human information processing capacity could also be helpful in establishing threshold in the metrics for the assuring database quality.

As we can see in figure 2, the process of defining and validating database metrics is evolutionary and iterative. As a result of the feedback metrics could be redefined of discarded depending of the theoretical, empirical or psychological validations.

4. Metrics for object-relational databases

4.1. First definition of our metrics

Taking into account the characteristics of an object-relational schema, we proposed three metrics:

Size of a Schema(SS)
We define the size of a schema as the sum of the size of the tables of the schema:

$$SS = \sum_{i=1}^{NT} TS_i$$ being NT the number of tables in the schema

We define the table size (TS) as the sum of the total size of the simple columns (TSSC) and the total size of the complex columns (TSCC) in the table:

$$TS = TSSC + TSCC$$

We consider that all simple columns have a size equal to one, and then the TSSC metric is equal to the number of simple columns in the table (NSA).

$$TSSC = NSA$$

And the TSCC is defined as the sum of each complex column size (CCS):

$$TSCC = \sum_{i=1}^{NCC} CCS_i$$ being NCC the number of complex columns in the table.

The value for CCS is obtained as the "size of the hierarchy (formed by the class and its ancestors)" above which the column is defined and may be defined as the sum of each class size in the hierarchy (SC):

$CCS = \sum_{i=1}^{NCH} SC_i$ being NCH the number of classes in the hierarchy.

The size of a class is defined as:

$$SC = \frac{SAC + SMC}{NHC}$$

being SAC the sum of the size attributes of the class, SMC the size methods of the class and NHC the number of hierarchies to which the class pertain.

The attributes of a class may also be simple or complex, then the SAC is defined as the sum of the simple attributes size (SAS, that have size equal to one like simple attributes) and the complex attributes size (CAS) in the class.

$$SAC = SAS + CAS$$

And the SMC is calculated with the version of the cyclomatic complexity given by [12]:

$SMC = \sum_{i=1}^{NMC} V_i(G)$ being NMC the number of methods in the class

Complexity of References Between Tables

In object-relational databases, other characteristics of relational databases are preserved, the most important of which is referential integrity. We propose to use two metrics for referential integrity (Calero et al., 1999):

- Depth of Relational Tree (DRT) is defined as the longest referential path between tables in the schema database
- Referential Degree (RD) is defined as the number of foreign keys in the schema database:

  $RD = \sum_{i=1}^{NT} NFK_i$ being NT the number of tables in the schema.

4.2. First empirical validation of our metrics

Once we had the metrics defined based on the specific characteristics of the object-relational databases, we made a proof with a small set of people for assuring that the metrics were correctly designed.

We designed an example with an object-relational table. This table had five columns, two of them were simple columns and three of them were complex columns (two of these three were defined above the same class hierarchy and the other column above another class hierarchy).

People must to calculate the size of each column and we recorded the time needed for each subject of the team. Finally we calculated the average time needed for each column and we observed that they weren't differences between the time expended to calculate the time of the simple columns but it was a big difference between the average of the time needed to calculate the size of the complex columns that shared the class hierarchy. When the persons calculated the size the first time they expended more time than the second time.

We thought about these results and we ended that perhaps it was more easy to calculate the value of the class hierarchy once they knew the hierarchy. So we decided to reflect this fact in the metric definition dividing the size of the class hierarchy between the number of columns that used it.

4.3. Debugging of the SS metric definition

The final definition of the SS metrics is:

We define the size of a schema in the same way:

$$SS = \sum_{i=1}^{NT} TS_i$$

and the table size (TS) also as the first time:

$$TS = TSSC + TSCC$$

Also the TSSC and the TSCC have the same expression:

$$TSSC = NSA$$

$$TSCC = \sum_{i=1}^{NCC} CCS_i$$

But now, the value for CCS is obtained as:

$$CCS = \frac{SHC}{NCU}$$

Being SHC the "size of the hierarchy (formed by the class and its antecessors)" above which the column is defined and NCU is the number of columns defined above this hierarchy. This expression is used to reflect the fact detected in the proof.

The SHC may be defined as the sum of each class size in the hierarchy (SC):

$$SHC = \sum_{i=1}^{NCH} SC_i$$ being NCH the number of classes in the hierarchy.

And the rest of the metrics have the same definition than before.

4.4. Example of our metrics

We present the values for the different metrics for the example presented in figure 1. Let us assume that all methods have a cylomatic complexity equal to 1 and all the large objects (LOBs) have also a size equal to one. In table 1, we present the value of each user defined type size.

Name Class	SAS	CAS	SAC	SMC	SC
address	4	0	4	0	4
employee	3	0	3	1	4

Table 1. Size values of the classes

The value for TSSC is given by the four simple columns of the table house. The value for TSCC is given by the four complex attributes with size equal to one (desc, front_view and document) plus the size of the complex column location plus the size of the complex columns address. So, we obtain the next values for the rest of metrics and finally for the table size:

$$CCS_{address} = \frac{4}{1} = 4 \qquad TSSC = 4$$

$$CCS_{employee} = \frac{4}{1} = 4 \qquad TSCC = 3 + 4 + 4 = 12$$

$$TS_{house} = 12 + 4 = 16 \qquad TS_{person} = 2$$

SS = 16 + 2 = 18

DRT=1 and RD=1

4.5. Formal verification of the metrics

Once the metric definition was made and refined we were able to made the formal verification of the metrics. In table 2 we present the results of the formal verification of our metrics in two formal frameworks: the formal framework proposed by Briand et al. (1996) as an example of axiomatic approach and the Zuse's formal framework (Zuse, 1998) which is based on measurement theory. In Piattini et al. (1998) and Calero et al. (1999) it is possible to find a complete explanation of Briand et al and Zuse formal frameworks and the formal verification of the metrics respectively.

	BRIAND ET AL(1996)	ZUSE(1998)
SS	SIZE	ABOVE THE ORDINAL
TS	SIZE	ABOVE THE ORDINAL
RD	COMPLEXITY	ABOVE THE ORDINAL
DRT	LENGTH	ABOVE THE ORDINAL

Table 2. Summary of metrics formal validation

4.6. Empirical validation

We prepared an experiment for validating empirically our metrics. In Piattini et al (2000) it is possible to find the complete experiment developed.

The experiment was developed in order to evaluate whether the proposed measures can be used as indicators for estimating the maintainability of an OR database. Five object-relational databases were used in this experiment with the average of 10 relations per database (ranging from 6 to 13). The people were given a form, which include for each table, a triplet of values to compute using the corresponding schema. These values are those of three measures TS, DRT and RD. Our idea is that to compute these measures, we need to understand the subschema (objects and relations) defined by the concerned table. A table (and then the corresponding subschema) is easy to understand if (almost) all the people find the right values of the metrics in a limited time (2 minutes per table).

To analyze the usefulness of the metrics proposed, we used two techniques: C4.5 (Quinlan, 1993), a machine learning algorithms and RoC (Ramoni and Sebastiani, 1999), a robust Bayesian classifier

From both techniques we find out that the table size metric (TS) is a good indicator for the maintainability of a table. The depth of the referential tree metric (DRT) is also presented as an indicator by C4.5 and the referential degree metric (RD) does not seem to have a real impact on the maintainability of a table.

5. Conclusions

Object-relational database management systems are replacing simpler relational ones. One of the main consequences of this change will be the stronger weight of the ORDBMSs in software systems maintainability.

It is necessary to dispose on metrics for controlling the complexity of this kind of databases. For obtaining good metrics is necessary to follow a set of steps. The method that we use for defining metrics is composed by the definition of the metrics, the formal verification, and the empirical validation of the metrics proposed. All these steps are related making a loop between the different steps. This loop means that a feedback could be necessary for obtaining good metrics.

In this paper we present the feedback, derived from an empirical validation, made in one of the proposed metrics for object-relational databases. From this empirical validation, we detected an error in the definition of a metric and its redefinition was necessary.

With this paper, we want to point up that the proposal of metrics for object-relational databases is not so easy that it seems and it is necessary to be careful in the definition.

Acknowledgement

This research is part of the MANTICA project, partially supported by the CICYT and the European Union (CICYT-1FD97-0168) and by the CALIDAT project carried out by Cronos Ibérica (supported by the Consejería de Educación de la Comunidad de Madrid, Nr. 09/0013/1999).

July/August, Nr. 4. Pp. 456-473.

(Briand et al., 1996) Briand, L.C., Morasca, S. And Basili, V. (1996). Property-based software engineering measurement. IEEE Transactions on Software Engineering, Vol.22(1). pp. 68-85.

(Calero et al., 1999) Calero, C., Piattini, M., Ruiz, F. and Polo, M. (1999).Validation of metrics for Object-Relational Databases, International Workshop on Quantitative Approaches in Object-Oriented Software Engineering (ECOOP99), (Lisbon ,Portugal. June 1999), 14-18.

(Calero et al., 2001) Calero, C., Piattini, M. and Genero, M (2001), Metrics for controlling database complexity, Accepted chapter for the book Developing quality complex database systems: practices, techniques and technologies.

(Cannan, 1999) Cannan, S.J. (1999), The New SQL Standard Good, Bad or Simply Ugly, Jornadas de Ingeniería del Software y Bases de Datos (JISBD99), Cáceres, Spain, November 1999.

(Cant et al., 1995) Cant, S., Jeffrey, D.R. and Henderson-Sellers, B., (1995), A conceptual model of cognitive complexity of elements of the programming process, Inf. Software Technolofy, 37(7), 351-362.

(Fenton, 1994) Fenton, N. (1994).Software Measurement: A Necessary Scientific Basis. IEEE Transactions on Software Engineering, 20(3): 199-206.

(Henderson-Sellers, 1996) Henderson-Sellers, B. (1996). Object-oriented Metrics - Measures of complexity. (Upper Saddle River, New Jersey). Prentice-Hall,

(Kitchenham et al., 1997) Kitchenham, B. and Stell, J.G. (1997). The danger of using axioms in software metrics. IEE Proc.-Softw. Eng. Vol.144, No 5-6, October-December, pp 279-285

(Klemola, 2000) Klemola, T. (2000) A cognitive model for complexity metrics, ECOOP´2000, Cannes, 13 June.

[illegible] Trans. on Software Engineering, 13 (6): 679-708.

(Morasca et al., 1997) Morasca, S. and Briand, L.C. (1997). Towards a Theoretical Framework for measuring software attributes. Proceeding of the Fourth International, Software Metrics Symposium, 119-126.

(Piattini et al., 1998) Piattini, M., Calero, C., Polo, M. and Ruiz, F. (1998). Maintainability in Object-Relational Databases. . Proc of The European Software Measurement Conference FESMA 98, Antwerp, May 6-8, Coombes, Van Huysduynen and Peeters (eds.). pp. 223-230.

(Piattini et al., 2000) Piattini, M., Calero, C., Sahraoui, H. and Lounis, H. (2000) An empirical study with object-relational database metrics, ECOOP´2000, Cannes, 13 June.

(Quinlan, 1993) Quinlan, J.R., C4.5: Programs for Machine Learning, (1993), Morgan Kaufmann Publishers.

(Ramoni et al., 1999) Ramoni, M. and Sebastiani, P. (1999). Bayesian methods for intelligent data analysis. In M. Berthold and D.J. Hand, editors, An Introduction to Intelligent Data Analysis, (New York). Springer.

(Siau, 1999) Siau, K. (1999). Information Modeling and Method Engineering: A

Psychological Perspective. Journal of Database Management 10 (4), 44-50.

(Stonebraker et al., 1999) Stonebraker, M. and Brown, P. (1999) Object-Relational DBMSs tracking the next great wave, (California), Morgan Kauffman Publishers.

(Weyuker, 1988) Weyuker, E.J. (1988). Evaluating software complexity measures. IEEE Transactions on Software Engineering Vol.14(9). pp. 1357-1365.

Whitmire, S.A. (1997), Object Oriented Design Measurement, Ed. Wiley.

(Zuse, 1998) Zuse, H. (1998). A Framework of Software Measurement. Berlin, Walter de Gruyter.