The European Software
Measurement Conference

FESMA 98

Business improvement through software measurement

Editors:
H. Coombes
M. Hooft van Huysduynen
B. Peeters

TECHNOLOGISCH INSTITUUT

ANTWERP, BELGIUM
May 6-8, 1998

Proceedings

The European Software Measurement Conference

FESMA 98

Business Improvement through Software Measurement

(A cvontinuation of ESCOM : European Software Control and Measurement Conference)

Antwerpen, May 6-8, 1998

Editors : H. Coombes, UK
M. Hooft van Huysduynen, NL
B. Peeters, B

organized by : Technologisch Instituut vzw (Technological Institute), Section on Software Metrics

Sponsored by : IFPUG, USA
Supported by the European Commission, DG III - Industry

ISBN 90-76019-03-7
D 1998/0277/02

CONTENTS

MAINTAINABILITY IN OBJECT-RELATIONAL DATABASES

Mario Piattini, Coral Calero, Macario Polo, Francisco Ruiz

Grupo ALARCOS
Departamento de Informática
University of Castilla-La Mancha
Ronda de Calatrava, 5
13071, Ciudad Real (Spain)
e-mail: {mpiattin, fruiz} @inf-cr.uclm.es
{coralc,mpolo}@caos.inf-cr.uclm.es

Keywords: Object-relational databases, software metrics, complexity, maintainability.

ABSTRACT

Databases have been demonstrated a tremendous productivity and an impressive economical impact. Nowadays "Third-generation" DBMSs are appearing into the market. This technology could arise some difficulties to its users because of the higher complexity of the database schema, which has been characterised by its simplicity in the classical relational model. We propose a metric suite for controlling object-relational database maintainability.

1.- INTRODUCTION

Since the seventies software engineers have been proposing all sorts of metrics for software products, processes and resources (Fenton, 1991). These metrics were defined in order to control some internal attributes of software elements (e.g. complexity, consistency, modularity) which influence external attributes (e.g. maintainability, testability, portability, understandability) that concern software development centres. It is well known that maintenance is the most important problem of software development, ranging between 67 and 90% of life-cycle costs (Frazer, 1992; McClure, 1992)

Unfortunately, almost all the metrics proposed since McCabe's cyclomatic number (McCabe, 1976) (by far the most referenced complexity metric) until now, focused on programme characteristics disregarding databases.

This neglection could be explained as databases have been until recently just "simple files/tables" which do not contribute too much to the complexity of the overall system.

But nowadays, with the appearance of the "third database generation", (Cattell, 1991), databases are becoming more and more complex, incorporating new data types, rules, generalisations, complex objects, etc. Last year we have assisted to the presentation of new "object-relational" products (Stonebraker, 1996) such as Informix/Illustra, Oracle 8 and DB2/2. And it is very probably that (after some years of delay) the new SQL3 standard will be published finally by late 1998. Figures about the DBMS market evolution confirm the prevision of an exponential grown of the object-relational DBMSs (Miranda, 1997).

In our opinion, the information systems developed with these new DBMSs will suffer of greater maintenance problems caused mainly by the own database schema. Two problems related with these new database systems are shortage of methodology and lacking of metrics considering their new functionalities.

We propose a metric suite which allows to control maintainability of object-relational database schema. Maintainability is considered to be influenced by understandability, modifiability and testability (Li and Cheng, 1987), which depend on size, length and complexity of database schemata.

In Section 2, we resume the main elements of an object-relational database schema. Then in Section 3 we define the metrics proposed. A formal description of the proposed metrics is presented in Section 4. Section 5 presents the conclusions and outlines the future work.

2.- ELEMENTS OF AN OBJECT-RELATIONAL DATABASE SCHEMA.

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

Also, generalisation hierarchies can be defined between classes (super and subclasses) and between tables (CREATE TABLE subtable UNDER supertable). Then, two types of associations can be established between tables, as shows figure 1.

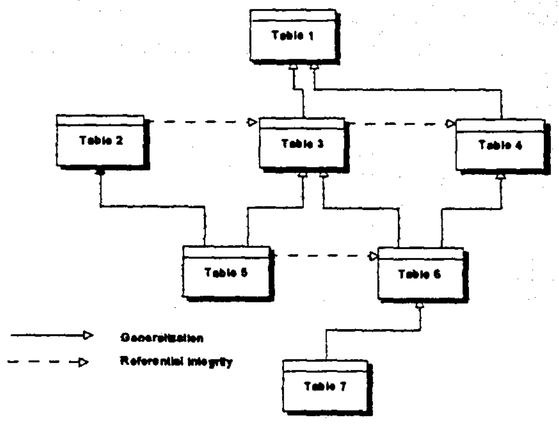

Figure 1. Example of an database object-relational schema

Object-relational databases support usually multiple inheritance (a subtable can be defined with more than one supertable).
Table attributes can be defined in a simple domain, e.g. char (25), or in a user-defined class as complex number or image, see figure 2. These classes can also be part of a generalisation lattice.

[1] In this first approximation domains and constraints are not considered for measures purposes.

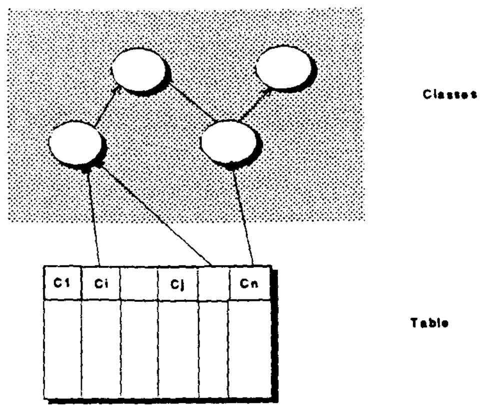

Figure 2. Example of complex column definition

3.- PROPOSED METRICS

The objective of this research is to propose a suite of metrics which can be obtained automatically from the database catalogue. This suite must be also sufficient but not excessive for practical purposes.

We consider, following the definitions of Briand et al. (1996), three types of metrics.

3.1.- Size metrics

Scheme size (SS). The scheme size (SS) is defined as the sum of the size of each table (TS) of the scheme.

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

Being NT the number of tables of the scheme.

Table size (TS). The table size (TS) is the sum of the size of its simple columns (SCS) which are considered with a size equal to one (so SCS will be the same to the number of simple columns) plus the size of its complex columns (CCS). The TS metric is characterised by the following expression:

$$TS = SCS + \sum_{i=1}^{NCC} CCS$$

Being NCC the number of complex columns in the table.

Complex Column Size (CCS). Complex columns of a table are defined on a class. For instance, the figure 2 shows that one column of a table may be defined on a class (which can belong to a class lattice), it shows too, that more than one column may be defined on the same class. So, we define the complex column size like the class size over it. The class size (CS) is defined recursively as:

$$CS = CSp + \sum_{i=1}^{NCF} CSi$$

Where NCF is the number of parents classes of the class.

Proper Class Size (CSp). To obtain the expression for the proper class size (CSp), we can consider the attributes and the methods included in the class. It's important to remember that attributes have less size than methods, so it's necessary to work with weighted values. CSp can be defined by adding its weighted attributes (ACS) plus its weighted methods (MCS):

$$CSp = (ACS + MCS)$$

We define ACS considering that the attributes have a size equal to one. So, ACS will be the same than the number of attributes per class (NOAC)

$$ACS = NOAC$$

Methods Class Size (MCS). MCS is defined using the version of the ciclomatic complexity of McCabe designed by Li and Henry (1993):

$$MCS = \sum_{i=1}^{nm} Vij(G)$$

Being nm the number of methods per class.

3.2.- <u>Length metrics</u>

The figure 1 shows that the tables may be related in two different ways. On one hand the generalisation relation (simple or multiple), on the other hand, the referential integrity (foreign key). For these characteristics we consider two length metrics.

The first one is the DIT metric of Chidamber and Kemerer (1994), it's named TDIT (Tables Depth Inheritance Tree) and it's defined:

$$TDIT = \textit{Depth maximum in the inheritance tree}$$

The second one it's based as well on Chidamber and Kemerer's metric but using the another kind of possible relation that can appear in a object-oriented scheme. It's named TDRT (Tables Depth Referential Tree) and it's defined like:

$$TDRT = \text{Maximum number of levels of referential integrity among tables}$$

For instance, in the figure 1, we have the next values for the TDIT and TDRT metrics: TDIT=3 and TDRT=2.

3.3.- Complexity metrics

We can define the referentiability degree (RD) of a scheme as the number of foreign keys (NFK) of all the tables in the schema (NT).

$$RD = \sum_{i=1}^{NT} NFK$$

In the example of the figure 1 we would have that: RD=3

4. FORMAL DESCRIPTION OF THE PROPOSED METRICS.

In this section we use the properties proposed by Briand et al. (1996) in order to characterise the metrics defined in the previous section.

4.1.- SS Metric

From a formal point of view, we consider that a relational scheme is composed by a set of elements which correspond to the set of the columns of the tables of the schema. Tables are considered to be the "modules" of the system.

A size metric is characterised by the following properties (Briand et al, 1996):

1. Nonnegativity. The size of a system is nonnegative.
2. Null value. The size of a system is null if it has no elements.
3. Module additivity. The size of a system is equal to the sum of the sizes of two of its modules such that any element of the system is an element of either a module or the other.

Proving these properties for the SS metric is easy:

1. Seeing the different expressions that compose the SS metric, it is impossible to obtain a negative value.
2. If we have no attributes then SCS=0 and CCS=0, so TS=0 and SS=0.
3. For module additivity, three possibilities must be study:

   Case 1. In the two modules there is a simple attribute.

   Case 2. In the two modules there is a complex attribute (of any type).

   Case 3. In one of the module there is a simple attribute and in the other there is a complex attribute (of any type).

   It's indifferent to study if there is only an attribute or several, so we study the simpler case.

In the first case, TS=SCS for the two modules, so TS will be the sum of both. In the second case, TS=CCS for the two modules, so TS will be the sum of both. In the last case, the first module gives a value for TS=SCS and the second gives a value for TS=CCS, TS will be the sum of both.

4.2.- TDIT and TDRT Metrics

For these metrics, we consider that a relational scheme is composed by a set of tables (elements) with their relations.
These relations may be of two types: inheritance when TDIT metric is analysed or referential integrity when TDRT metric is analysed.

For length metrics, the properties given by Briand et al. (1996) are:

1. Nonnegativity. The length of a system is nonnegative.
2. Null value. The size of a system is null if has no elements.
3. Nonincreasing monotonicity for connected components. Let S be a system and m be a module of S such that m is represented by a connected component of the graph representing S. Adding relationships between elements of S does not increase the length of S.
4. Nondecreasing monotonicity for nonconnected components. Let S be a system and m1 and m2 be two modules of S such that m1 and m2 are represented by two separate connected components of the graph representing S. Adding relationships from elements of m1 to elements of m2 does not decrease the length of S.
5. Disjoint modules. The length of a system made of two disjoint modules m1, m2 is equal to the maximum of the lengths of m1 and m2.

The demonstration of these properties for the TDIT and TDRT metrics is the following:

1. The depth of a tree never can be negative.
2. If we have no attributes (E=∅), we have no tree, we have no depth and TDIT=TDRT=0.
3. If we add relationships between elements of a tree (tables) the depth does not vary. This will be only possible by adding elements to the tree.
4. If we add relationships between elements of two trees we cannot decrease the depth of the resulting tree because the depth of the first tree stays equal and we add a relation which has a depth 1 as minimum (if it's a leaf tree).
5. The depth of a tree is given by the component which has more levels from the root to the leaves.

So, we can conclude that TDIT and TDRT are length metrics.

4.4.- RD Metric

To demonstrate that RD is a complexity metrics, we prove that verifies the properties given by Briand et al. (1996) for this kind of metrics:

1. Nonnegativity. The complexity of a system is nonnegative.
2. Null value. The complexity of a system is null if it has no relations.
3. Symmetry. The complexity of a system does not depend on the convention chosen to represent the relationships between its elements.
4. Module Monotonicity. The complexity of a system is no less than the sum of the complexities of any two of its modules with no relationships in common.
5. Disjoint Module Additivity. The complexity of a system composed of two disjoints modules is equal to the sum of the complexities of the two modules

RD metric verifies these properties because:

1. Seeing the expression that composes the RD metric, it is impossible to obtain a negative value.
2. If there is no relations there can be referential integrity, so RD = 0.
3. The definition of RD is the same disregarding the direction of the reference.
4. If the modules are no disjoint, this means that between elements of both modules there is a relation of referential integrity, so RD never decrease.
5. Every module will have a value for NFK. When modules are disjoint neither foreign key nor a table will be common to both modules, so the result of RD of the system will be the sum of the NFK of the two modules, and so RD will be the sum of the RD of the modules.

5.- CONCLUSIONS AND FUTURE WORK

We have presented a first approximation for measuring object-relational databases. Five different metrics are defined and described using a formal framework.

However, the framework used is not the only one (see for example Weyuker 1988) or Fenton (1991)) and it is not generally accepted (Kitchenham and Stell, 1997). So we must validate these metrics using other axioms definition.

Also empirical validation is being carried out, not only to prove metrics validity, but also to give some limits which can be useful for database designers.

Adaptation of these metrics to SQL3 is also a current work, since it can improve the metric suite, (Churcher y Shepperd, 1995) .

Finally, an automatic tool is being developed in order to automate the metric gathering for ORACLE 8 DBMS.

ACKNOWLEDGEMENTS

This research is part of the MANTICA project (a proposal to CICYT-FEDER) and is partially supported by the MANTEMA project carried out by ATOS ODS, S.A. (ATYCA Dirección General de Tecnología y Seguridad Industrial del Ministerio de Industria y Energía).

Measurement". IEEE Trans...
January, 1996.

- Cattell, R.G.G. "What are next-generation database systems?". CACM, Vol. 34, Nº 10, 1991, pp. 31-33.
- Chidamber, S. and Kemerer, C. "A metrics suite for object-oriented design", *IEEE Trans. Software Eng.*, vol. 20,no. 6, June, 1994, pp. 476-493.
- Churcher, N.J. y Shepperd, M.J. "Comments on "A Metrics Suite for Object-Oriented Design". EN: IEEE Trans. on Software Engineering Vol. 21, Nº 3, 1995, pp. 263-265.
- Fenton, N. "Software Metrics: A Rigorous Approach". London, Chapman & Hall. 1991.
- Frazer, A. "Reverse engineering-hype, hope or here?". In: P.A.V. Hall, Software Resue and Reverse Engineering in Practice. Chapman & Hall. 1992.
- Henderson-Sellers, B. "Object-Oriented Metrics". The object-oriented Series. Prentice Hall. 1996
- Li, H.F. and Chen, W.K. "An empirical study of software metrics". IEEE Trans. on Software Engineering SE-13 (6), 1987, 679-708.
- Li, W. and Henry, S. "Object-Oriented metrics that predicts maintainability". *J.Sys.Software*, 23, 1993, pp. 111-122.
- McCabe, T.J. " A complexity measure". IEEE Trans. Software Engineering Vol 2, Nº 5, 1976, pp. 308-320.
- McClure, C. "The Three R's of Software Automation: Re-engineeirng, Repository, Reusability". Englewood Cliffs: Prentice-Hall, 1992.
- Miranda, S "Obect Relational Data Models of the Future". En: Proc. of the Third Basque International Workshop on Information Technology. BIWIT'97". July 2-4, 1997, Biarritz, Francia, IEEE Computer Society.
- Stonebraker, M. "Object-Relational Databases. The next-Wave". Morgan-Kauffmann, 1996..
- Weyuker, E.J. "Evaluating software complexity measures". IEEE Transactions on Software Engineering Vol. 14, Nº 9, pp. 1357-1365, Sept.1988.