VOLUME 11 NUMBER 3 2002 ISSN# 1535-6698

NOVA

International Journal of Computer Research

SUBSCRIPTION INFORMATION
2001. Volume 1 (4 issues)
Quarterly $300
Institutions $45 Authors and Referees $45
Individuals $45 Students $45

Nova Science Publishers, Inc.
400 Oser Avenue, Suite 1600
Hauppauge, New York 11788
Tele. 631-231-7269
Fax 631-231-8175
E-Mail: Novascience@earthlink.net
Web Site: http://www.novapublishers.com

Production Staff: Publisher: Frank Columbus; Vice President: Nadya Gotsiridze; Senior Editors: Susan Boriotti and Donna Dennis; Coordinating Editor: Tatiana Shohov; Office Manager: Annette Hellinger; Graphics: Wanda Serrano; Book Production: Matthew Kozlowki, Jonathan Rose and Jennifer Vogt; Circulation: Cathy DeGregory, Ave Maria Gonzalez and Raheem Miller; Communications and Acquisitions: Serge P. Shohov

International Journal of Computer Research

Volume 11, Number 3, 2002

TABLE OF CONTENTS

Nova Science Publishers, Inc.
New York
2002

[redacted]

Database Metrics Definition and Automation

Coral Calero, Manuel A. Serrano, Mario Piattini, and Marcela Genero
Grupo ALARCOS - E.S.Informática.
Universidad de Castilla – La Mancha
Ronda Calatrava, 5 13071 Ciudad Real
Tfno: +34926295300 Ext:3715
Fax: +34926295354
SPAIN
E-mail: {ccalero, mserrano, mpiattin, mgenero}@inf-cr.uclm.es

Abstract

Software measurement is an effective means to manage software development and maintenance projects. In the past decades a huge amount of software metrics has been proposed, but primarily focused on programs. Metrics for databases have been neglected, mainly because databases have developed a secondary role in Information Systems (IS) infrastructure until a few years ago. But nowadays, databases are the core of IS, influencing considerably their complexity. This paper proposes a metrics suite for measuring relational database complexity, apply them to ORACLE and presents the tool we are developing to automate the metrics calculation and visualization.

Key-Words: - Software metrics, database maintainability, ORACLE.

1 Introduction

In the last years we have beheld an extraordinary diffusion of the relational databases, that have been installed on macrocomputers an also on personal computers. The databases, especially the relational databases, have became the core of the most relevant Information Systems. However, it design is a long and hard task ([9]), although it is based on solid mathematical fundaments derived from the definition of the relational model ([7]).

Most of the methodologies in designing relational databases ([1], [14], [8], [20]) are divided into three steps: conceptual modelling, logic design and physical design; applying a set of rules that allow the transformation of a conceptual schema to a relational schema and that guarantee the "quality" of the relational schema applying the normalisation theory. Although this theory is very useful, it is necessary to develop new metrics to evaluate the quality of a relational database. In fact, currently, it is recognised that software measurement is an effective form to understand, monitor, control, predict and improve the development and maintenance of software projects ([2]). Also, they can help the practitioners and the investigators to make the best decisions ([23]), helping the designer to choose among alternative database schemas.

Unfortunately, most of the metrics ever since McCabe proposed the cyclomatic complexity ([18]) to the present time, have been centred in the program characteristics, disregarding databases ([24]). Nevertheless, design metrics for data aspects is important since their size and nature contribute to many aspects of a information system like the amount of effort to develop ([19]).

One of the most important aspects relative to quality, is maintainability ([16]) which depends on three factors: understandability, modifiability and testeability, which are influenced by complexity ([17]). However, a general complexity measure is the impossible holy grail ([11]). In [15], the author distinguishes three types of complexity: representational, computational and psychological. The last one is composed by the problem complexity, the human cognitive factors and product complexity, in which we are currently investigating.

In this paper we propose a set of metrics for measuring the relational database schemas complexity: NA (number of attributes), RD (Referential Degree), DRT (Depth Referential Tree) and COS (Cohesion of the schema).

Although generic metrics have been proposed for relational databases, it is necessary to adapt them to specific DBMS characteristics since it can improve our metric suite ([6]), in order to obtain appropriate values for the metrics.

In this way we have selected ORACLE to apply our metrics due to the importance that this DBMS have in real world and because it owns a leading technology.

Section 2 presents the proposed metrics. Section 3 presents its formal validation. The application of the metrics for ORACLE is given in section 4 and the conclusions and future work come in the last section.

2 Metrics for Relational Databases

Since the proposal by Dr. Codd of the relational model in the late sixties ([7]), the database field has been extensively researched and relational database products have developed a very important industry.

The only indicator used to measure the quality of relational database has been the normalisation theory, upon which ([13]) propose to obtain a normalisation ratio.

We propose a set of metrics that can be divided in table oriented metrics and schema oriented metrics (two of them, NA and RD, are table oriented and schema oriented).

- As table oriented metrics we propose:

Number of attributes (NA(A))
NA(A) is the number of attributes of the table A.

Referential Degree (RD(A))
RD(A) is the number of foreign keys in the table A.

- As schema oriented metrics we propose:

Number of attributes (NA)
NA is the number of attributes in all the tables of the schema.

Referential Degree (RD)
The RD metric is defined as the number of foreign keys in the schema.

Depth Referential Tree (DRT)
DRT is defined as the length of the longest referential path in the database schema. The cycles are only considered once.

Cohesion of the schema (COS)
COS is defined as the sum of the square of the number of tables per unrelated subgraph in the database:

$COS = \sum_{i=1}^{\lvert US \rvert} NTUSi^2$	|US| number of unrelated subgraphs $NTUS_i$ number of tables in the unrelated subgraph "i"

3 Applying Metrics to ORACLE Database

In this section we present the way to calculate the value of our metrics when working with ORACLE databases. In table 1 we present the results. The first column represents the metric calculates and the second column presents the SQL code associated for obtaining the value for the metric.

Table 1

	METRIC	SQL CODE
TABLE ORIENTED	NA(A)	SELECT ATC.TABLE_NAME, COUNT(ATC.COLUMN_NAME) FROM ALL_TAB_COLUMNS ATC, ALL_TABLES AT WHERE ATC.TABLE_NAME = AT.TABLE_NAME AND AT.OWNER = 'SYS' GROUP BY ATC.TABLE_NAME; Where : ALL_TAB_COLUMNS is a view that lists the columns of all tables, views, and clusters accessible to the user. ALL_TABLES is a view that contains descriptions of relational tables accessible to the user. SYS is owner the database's data dictionary .
	RD(A)	SELECT AC.TABLE_NAME, COUNT(AC.CONSTRAINT_NAME) FROM ALL_CONSTRAINTS AC, ALL_TABLES AT WHERE AC.CONSTRAINT_TYPE = 'R' AND AC.TABLE_NAME = AT.TABLE_NAME AND AT.OWNER = 'SYS' GROUP BY AC.TABLE_NAME; Where: ALL_CONSTRAINTS is a view that lists constraint definitions on accessible tables. ALL_TABLES is a view that contains descriptions of relational tables accessible to the user. SYS is owner the database's data dictionary . R is the type of constraint definition for referential integrity.

METRIC ORIENTED	**NA**	SELECT COUNT(ATC.COLUMN_NAME) FROM ALL_TAB_COLUMNS ATC, ALL_TABLES AT WHERE ATC.TABLE_NAME = AT.TABLE_NAME AND AT.OWNER = 'SYS'; Where : ALL_TAB_COLUMNS is a view that lists the columns of all tables, views, and clusters accessible to the user. ALL_TABLES is a view that contains descriptions of relational tables accessible to the user. SYS is owner of the database's data dictionary .
	RD	SELECT COUNT(*) FROM ALL_CONSTRAINTS AC, ALL_TABLES AT WHERE AC.CONSTRAINT_TYPE = 'R' AND AC.TABLE_NAME = AT.TABLE_NAME AND AT.OWNER = 'SYS'; Where: ALL_CONSTRAINTS is a view that lists constraint definitions on accessible tables. ALL_TABLES is a view that contains descriptions of relational tables accessible to the user. SYS is owner the database's data dictionary . R is the type of constraint definition for referential integrity.
	DRT and COS	In order to calculate the values for these metrics, we must export the database schema to the tool metadatabase. With the information in the metadatabase and using graphs theory, the application calculates de metrics values.

Table 1 continued

4 An overview of MANTICA Tool

In order to have an effective metric suite we need a methodological support, but is also important to provide us with metrics tools that allow us to calculate the metrics values in a practical, efficient and exact way. This tool must automate the acquisition, presentation and analysis of the metrics values.

Having a metric tool we can obtain some advantages:

- We can calculate the metrics values with little effort.
- We can also minimize the errors calculating the metrics values.
- The tool can calculate the metrics values without our contribution, so we can focus on the analysis phase.

A tool for collecting and visualizing all these metrics has been developed. Our tool is prepared to measure different kinds of databases schemas. In order to do it, we designed a generic metadatabase [5], where al the elements must be described depending on the type of the database the tool is measuring.. The user interface is also a very important issue in a metric tool like MANTICA, due to the huge amount of data and statistical analysis which must be visualised (see figure 1).

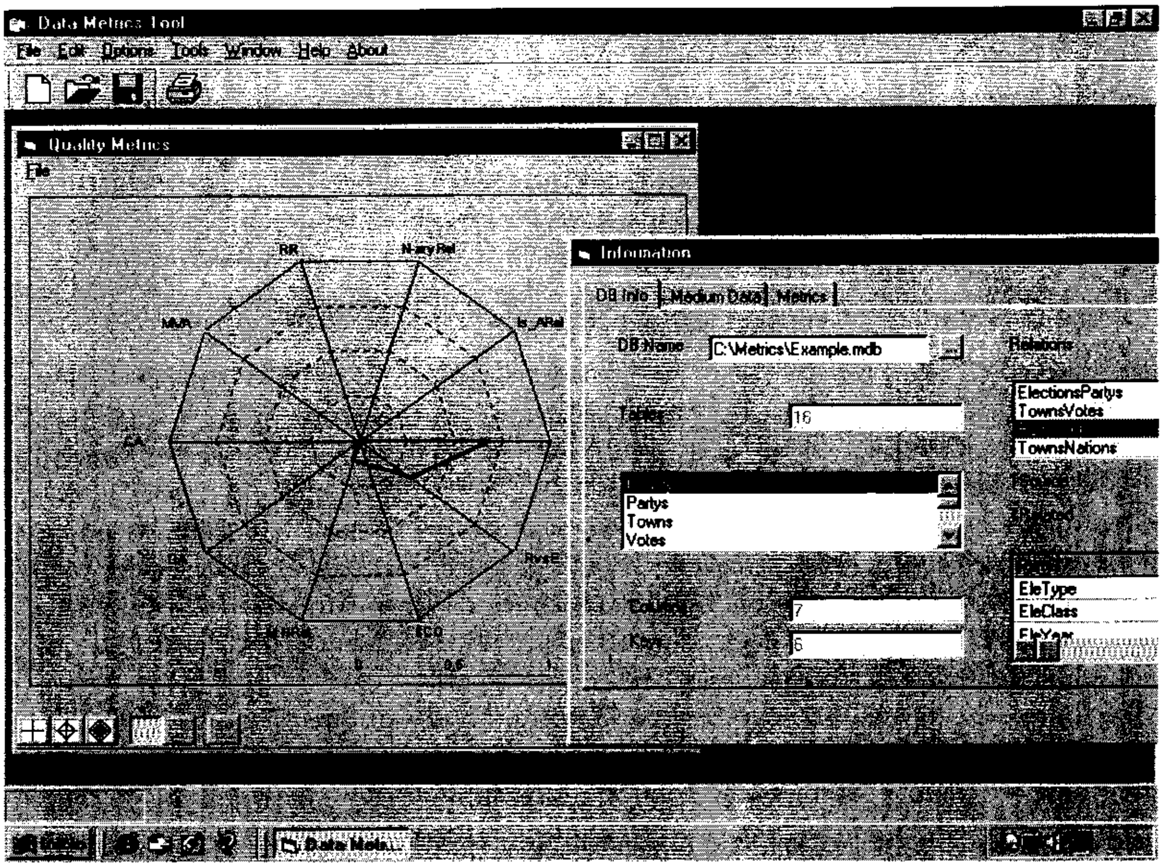

Figure 1. MANTICA interface

5 Conclusions and Future Work

More research is needed in software measurement ([21]), both from theoretical and from practical points of view ([12]). We think it is very interesting to dispose on metrics for relational databases. Also, it is necessary to adapt the metrics to specific DBMS.

We have presented a set of metrics to measure the database schema complexity in order to have a mechanism to improve and assure the quality of databases. The formal validation of the proposed metrics was made ([3]) following the formal framework presented in [25]. Also we have carried out an empirical validation with the referential integrity related metrics ([4]) obtaining that the number of foreign keys (RD) is a solid indicator of the database schema understandability and the length of the referential path (DRT) can modulates the RD metric effects.

We have also presented an overview of our tool for automatic metrics collection and visualization.

Our work is complemented with others metrics for active ([10]) and object-relational databases ([22]). We are also adapting our metrics to several DBMS that followed the SQL2 model, simplifying the measures to be taken. In this way, we can give a more precise guide to the metrics usage ([6]).

Acknowledgment

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

References

[1] Batini, C., Ceri, S. y Navathe, S., (1992), *Conceptual database design. An entity relationship approach.* Benjamin Cummings Publishing Company.

[2] Briand, L., Morasca, S. and Basili, V. "Property-Based Software Engineering Measurement". *IEEE Transactions on software Engineering,* Vol. 22, No 1, January, 1996.

[3] Calero, C., Piattini, M., Genero, M., Serrano, M. y Caballero, I. (2000). Metrics for Relational Databases Maintainability, United Kingdom Academy of Information Systems, UKAIS 2000, Cardiff, UK. pp. 109-119. ISBN 0-07-7097556. McGraw-Hill.

[4] Calero, C., Piattini, M., Polo, M. y Ruiz, F. (1999).Validating referential integrity as a database quality metric. *ICEIS* 99, 27-31 March. Portugal.

[5] Calero, C., Piattini, M., Serrano, M.A., Genero, M. (2000). Appliying metrics to Oracle Databases. *EOUG 2000,* Madrid, Spain. 21-23 June.

[6] Churcher, N.J. y Shepperd, M.J. (1995), Comments on "A Metrics Suite for Object-Oriented Design. *IEEE Trans. on Software Engineering* Vol. 21, Nº 3, 1995, pp. 263-265.

[7] Codd, E.F. (1970). A Relational Model of Data for Larged Shared Data Banks. *CACM,* 13 (6), 377-387.

[8] Connolly, T., Begg, C. y Strachan, A., (1998), *Database systems,* 2ª Ed. Addison-Wesley

[9] De Miguel, A., Piattini, M., (1993), *Concepción y diseño de bases de datos relacionales,* Ed. Addison-Wesley

[10] Díaz, O., Piattini. M. y Calero, C. (2001). Measuring active databases maintainability. Accepted for publication in *Information Systems Journal*

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

[12] Glass, R. (1996). The Relationship Between Theory and Practice in Software Engineering. *IEEE Software,* November, 39 (11), 11-13.

[13] Gray R.H.M., Carey B.N., McGlynn N.A., Pengelly A.D., (1991), Design metrics for database systems. *BT Technology* J, Vol 9, 4 Oct, 69-79

[14] Hansen, G.W. y Hansen J.V., (1997), *Diseño y administración de bases de datos,* 2ª Ed. Ed. Prentice-Hall

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

[16] ISO, (1994). *Software Product Evaluation-Quality Characteristics and Guidelines for their Use.* ISO/IEC Standard 9126, Geneva.

[17] Li, H.F. y Chen, W.K. (1987). An empirical study of software metrics. *IEEE Trans. on Software Engineering,* 13 (6): 679-708.

[18] McCabe, T.J. (1976). A complexity measure. *IEEE Trans. Software Engineering* 2(5), pp.308-320.

[19] MacDonell, S.G., Shepperd, M.J. y Sallis, P.J. (1997). Metrics for Database Systems: An Empirical Study. *Proc. Fourth International Software Metrics Symposium – Metrics'97*, Albuquerque. IEEE Computer Society, pp. 99-107.

[20] McFadden, F.R., Hoffer, J.A. y Prescott, M.B., (1999), *Modern database management*, 5ª Ed. Addison-Wesley

[21] Neil, M. (1994) Measurement as an Alternative to Bureaucracy for the Achievement of Software Quality. *Software Quality Journal* 3 (2), 65-78.

[22] 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.), 223-230.

[23] Pfleeger, S. L. (1997). Assessing Software Measurement. *IEEE Software*. March/April, pp. 25-26.

[24] Sneed, H.M. y Foshag, O. (1998). Measuring Legacy Database Structures. Proc of The European Software Measurement Conference FESMA 98, Antwerp, May 6-8, Coombes, Van Huysduynen and Peeters (eds.), 199-211.

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