INFORMATION SYSTEMS – RESEARCH, TEACHING AN PRACTICE

Proceedings of the 5th UKAIS Conference
University of Wales Institute, Cardiff
26 – 28 April 2000

McGraw Hill

Published by McGRAW-HILL Publishing Company
Shoppenhangers Road, Maidenhead, Berkshire, SL6 2QL, England
Tel 01628 502500; Fax 01628 770224

British Library Cataloguing in Publication Data
The CIP data of this title is available from the British Library

Library of Congress Cataloging-in-Publication Data
The CIP data of this title is available from the Library of Congress, Washington DC, USA

ISBN 0 07 709755 6

Printed and bound in Great Britain at the Basingstoke Press (75) Ltd, Basingstoke

PREFACE

Paul Beynon-Davies
Michael D. Williams
Ian Beeson

1. UKAIS2000

CROESO I GYMRU, CROESO I'R CYNHADLEDD UKAIS.
WELCOME TO WALES, WELCOME TO THE UKAIS CONFERENCE.

This years' UKAIS conference is held at the Cyncoed campus of the University of Wales Institute, Cardiff. The site was chosen in a selection process conducted by the Wales and West Regional Group of the UKAIS. The group is hosting the conference this year.

We feel that Cardiff offers an outstanding venue for the conference and welcome the fact that it is the first time the conference has been held outside England. We have attempted to formulate a varied and interesting conference programme that we hope everyone will enjoy.

2. REVIEW PROCESS

A grand total of 95 papers were submitted to the conference this year. Each paper was sent for blind review to two referees. The programme committee of the conference this year decided to introduce three classes of papers with the aim of encouraging inclusion amongst IS academics. In total 64 full papers, 9 short papers and 12 discussion papers were accepted.

3. PLENARY SPEAKERS, PANELS AND TUTORIALS

We have been lucky in getting a number of eminent persons from IS academia to speak at this years conference, indicating the increasing role that the conference plays in the IS calendar. The conference also includes a number of panel sessions and one tutorial session on the interesting subject of open source software.

4. PAPER STREAMS

The papers were assigned to streams on the basis of the UKAIS sub-areas contained in the definition for information systems published by the Academy. Papers were also categorised in terms of being either research, practice or teaching papers.

5. LIST OF REVIEWERS

Thanks to the following persons who acted as reviewers of the papers submitted this year.

Alison Adam, UMIST
David Avison, University of Southampton
Yehuda Baruch, University of East Anglia
Ian Beeson, University of the West of England
Frances Bell, University of Salford
Paul Beynon-Davies, University of Glamorgan
Michael Brittain, University of Wellington (NZ)
Laurence Brooks, University of York
Tom Gough, University of Leeds
Ray Hackney, Manchester Metropolitan University
Guy Fitzgerald, Brunel University
Donal Flynn, UMIST
Sue Fowell, University of New South Wales (Australia)
Matthew Jones, University of Cambridge
Chris Kimble, University of York
Lydia Lau, University of Leeds
Margi Levy, University of Warwick
Feng Li, University of Strathclyde
Michael Lloyd-Williams, Manchester Metropolitan
Nathalie Mitev, London School of Economics
Philip Powell, University of London
Hugh Preston, University of Wales, Aberystwyth
Steve Probert, Cranfield University
Nancy Russo, Northern Illinois University (USA)
Carsten Sorensen, London School of Economics
Frank Stowell, De Montfort University
Richard Vidgen, University of Bath
David Wainwright, University of Durham
Edgar Whitley, London School of Economics
Bernard Williams, University of East Anglia
David Wilson, University of London
Brent Work, University of Surrey
Gareth Griffiths (Manchester Metropolitan University)

6. MEMBERS OF THE UKAIS BOARD

Our thanks also to the UKAIS board for their support:

Laurence Brooks (York) - laurence.brooks@cs.york.ac.uk
Ray Hackney (Manchester Metropolitan) - r.hackney@mmu.ac.uk
Philip Powell (Newsletter Editor: Bath) - mnspp@management.bath.ac.uk
Feroz Siddiqui (Student Representative) - feroz.siddiqui@brunel.ac.uk
Steve Smithson (Treasurer:LSE) - s.smithson@lse.ac.uk
Frank Stowell (President: De Montfort) - fstowell@dmu.ac.uk
John Ward (Membership Secretary: Cranfield): j.m.ward@cranfield.ac.uk
Trevor Wood-Harper (Salford) - A.T.Wood-Harper@salford.ac.uk

TABLE OF CONTENTS

Short Papers

Discussion Papers

Panels & Tutorials

CHAPTER 13
METRICS FOR RELATIONAL DATABASES MAINTAINABILITY

Coral Calero, Mario Piattini, Marcela Genero, Manuel Serrano, Ismael Caballero
University of Castilla-La Mancha

Abstract.
Software measurement is widely recognised as an effective means to understand, control, predict and improve software development and maintenance projects. In the last decades a huge amount of software metrics has been proposed, but they focused primarily 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 maintenance. Maintainability is achieved by three factors: understandability, modifiability and testability, which in turn are influenced by complexity. This paper proposes different metrics for measuring relational database complexity. These measures are characterised using the measurement theory, particularly the formal framework put forward by Zuse (Zuse, 1998). Some empirical experiments were carried out in order to validate the proposed metrics.

1. INTRODUCTION

Metrics are useful mechanisms for the improvement of the quality of software products, specially maintenance which is the most important problem of software development, ranging between 60 and 90 percent of life-cycle costs (Card and Glass, 1990) (Kim and March, 1995). Software measurement is widely recognised as an effective means to understand, monitor, control, predict and improve software development and maintenance projects (Briand et al, 1996). Measurement is used not only for understanding, controlling, and improving development, but also for determining the best ways to help practitioners and researchers (Pfleeger, 1997).

Software engineers have been putting forward huge quantities of metrics for software products, processes and resources (Melton, 1996) (Fenton and Pfleeger, 1997). Unfortunately, almost all the metrics proposed since McCabe's cyclomatic number (McCabe, 1976) until nowadays, have focused on program characteristics disregarding databases. As far as databases are concerned, metrics have been used for comparing data models rather than the schemata themselves. Several authors (Juhn and Naumann, 1985) (Jarvenpaa and Machesky 1986) (Shoval and Even-Chaime, 1987) (Batra et al, 1990) [20], (Rossi and Brinkkemper, 1996) have compared the best well-known models such as E/R, NIAM and the relational ones using different metrics. Although we think this work is interesting, metrics for comparing schemata are needed mainly for practical purposes, such as choosing between different design alternatives or giving designers limit values for certain characteristics (analogously to value 10 for Mc Cabe´s complexity of programs).

Design metrics for data aspects are important because the size of data and their nature contribute to many aspects of a system like the amount of effort to develop. Then measure data can potentially help control and predict aspects of the data model during the software development process (MacDonell, Shepperd and Sallis, 1997).

However, as Sneed and Foshag remark: "metrics for databases have been neglected in the metric community"(Sneed and Foshag, 1998). This neglect could be explained as, until recently, databases have developed just a secondary role with minor contributions to the complexity of the overall system. Nowadays, databases have been introduced in most of the important IS, becoming their essential core. For this reason, we think it is very important to measure databases and understand their contribution to the overall IS maintainability.

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

Two different measures related with referential integrity are proposed in this paper for relational database schemata complexity: Depth of Referential Tree (DRT) and Referential degree (RD). These measures are characterised using measurement theory, particularly the formal framework proposed by Zuse [37].

Section 2 presents the different measures proposed. We give a brief introduction to the Zuse's framework in section 3, using it to characterise the metrics. Section 4 presents the experiments performed with the referential integrity related metrics. The conclusions and suggestions for future work are included in the last section.

2. MEASURES PROPOSED FOR RELATIONAL DATABASES

Since the Dr. Codd's proposal of the relational model in the late sixties (Codd, 1970) 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 databases has been the normalisation theory, upon which (Gray, Carey, McGlynn, Pengelly, 1991) propose to obtain a normalisation ratio. However, foreign key is one of the main concepts of the relational model. It can be defined as follows (Date, 1995): Let R2 be a base relation. Then a foreign key in R2 is a subset of the set of attributes of R2, say FK, such that:

1. There exists a base relation R1 (R1 and R2 not necessarily distinct) with a candidate key CK, and
2. For all time, each value of FK in the current value of R2 is identical to the value of CK in some tuple in the current value of R1.

Related with the foreign key concept, the relational model includes the referential integrity rule: the database must not contain any unmatched foreign key values (Date, 1995).

2.1 Measures definition

Based on the referential integrity, we propose two different metrics:

1. Depth Referential Tree (DRT). DRT is defined as the length of the longest referential path in the database schema and DRT(A) is the length of the longest referential path in the table A. The cycles are considered only once.
2. Referential Degree (RD). The RD metric is defined as the number of foreign keys in the schema and RD(A) is the number of foreign keys in the table A.

As a example, if we consider the following schema:

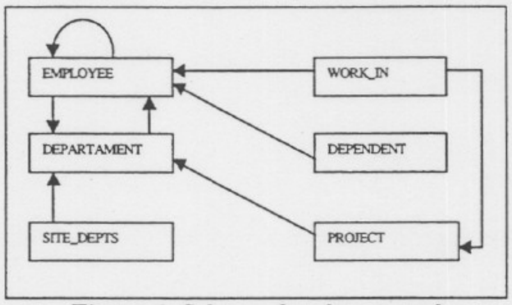

Figure 1. Schema for the example

Derived from the next example taken from (Elmasri and Navathe, 1997):

```
CREATE TABLE EMPLOYEE(
        NAMEP      VARCHAR(15)  NOT NULL,
        INIC       CHAR,
        SURNAME    VARCHAR(15)  NOT NULL,
        NSS        CHAR(9)      NOT NULL,
        DATEN      DATE,
        ADDRESS    VARCHAR(30),
        SEX        CHAR,
        SALARY     DECIMAL(10,2),
        NSSSUPER   CHAR(9),
        ND         INT          NOT NULL,
  CONSTRAINT CLPEMP
        PRIMARY KEY (NSS),
  CONSTRAINT CLESUPEREMP
        FOREIGN KEY (NSSSUPER) REFERENCES
EMPLOYEE(NSS)
        ON DELETE SET NULL ON UPDATE CASCADE,
  CONSTRAINT CLEDEPTOEMP
        FOREIGN KEY (ND) REFERENCES DEPARTAMENT
(NUMBERD)
        ON DELETE SET DEFAULT ON UPDATE CASCADE);

CREATE TABLE DEPARTAMENT(
        NAMED      VARCHAR(15)  NOT NULL,
        NUMBERD    INT          NOT NULL,
        NSSGTE     CHAR(9)      NOT NULL,
        DATEINICGTE DATE,
  CONSTRAINT CLPDEPTO
        PRIMARY KEY (NUMBERD),
  CONSTRAINT CLSDEPTO
        UNIQUE(NOMBRED),
  CONSTRAINT CLEGTEDEPTO
        FOREIGN KEY (NSSGTE) REFERENCES EMPLOYEE
(NSS)
        ON DELETE SET DEFAULT ON UPDATE CASCADE);

CREATE TABLE SITE_DEPTS(
        NUMBERD INT                  NOT NULL,
        SITED        VARCHAR(15)     NOT NULL,
  PRIMARY KEY (NUMBERD, SITED),
  FOREIGN KEY (NUMBERD) REFERENCES DEPARTAMENT
(NUMBERD)
        ON DELETE CASCADE ON UPDATE CASCADE);


CREATE TABLE PROJECT(
        NAMEPR       VARCHAR(15)     NOT NULL,
        NUMBERPR   INT               NOT NULL,
        SITEPR       VARCHAR(15),
        NUMD         INT             NOT NULL,
  PRIMARY KEY (NUMBERP),
  UNIQUE (NAMEP),
  FOREIGN KEY (NUMD) REFERENCES DEPARTAMENT
(NUMBERD));

CREATE TABLE WORK_IN(
        NSSE       CHAR(9)           NOT NULL,
        NUMP       INT               NOT NULL,
        HOURS      DECIMAL(3,1)      NOT NULL,
  PRIMARY KEY (NSSE, NUMP),
  FOREIGN KEY (NSSE) REFERENCES EMPLOYEE (NSS),
  FOREIGN KEY (NUMP) REFERENCES PROJECT (NUMBERP));

CREATE TABLE DEPENDENT(
        NSSE            CHAR(9)          NOT NULL
        NAME_DEPEND   VARCHAR(15)        NOT NULL,
        SEX             CHAR,
        DATEAN          DATE,
        RELATION       VARCHAR(8),
  PRIMARY KEY (NSSE, NAME_DEPEND),
  FOREIGN KEY (NSSE) REFERENCES EMPLOYEE (NSS));
```

We obtain the following values for the metrics (table 1):

	DRT	*RD*
EMPLOYEE	3	2
DEPARTAMENT	3	1
SITE_DEPTS	4	1
PROJECT	4	1
WORK_IN	5	2
DEPENDENT	4	1
SCHEMA	5	8

Table 1. Values of the metrics in the example

3. AN INTRODUCTION TO ZUSE MEASUREMENT FRAMEWORK

Several frameworks for measure characterisation have been proposed (Briand, Morasca and Basili, 1996) (Weyuker, 1988). In this paper we will follow the formal framework of [37] in order to describe the properties of the metrics defined above. This framework is based on an extension of the classical measurement theory, which gives a sound basis of software measures, their validation and criteria for measurement scales.

Dr. Zuse describes measurement as a detour, "a *necessary one because humans mostly are not able to make clear and objective decisions or judgements*" ([37]). Measuring is more than producing numbers, it is the combination of empirical entities with numerical entities. People are interested in establishing "empirical relations" between objects, such as "higher than" or "equally high or higher than". These empirical relations will be indicated by the symbols "• >" and "• >=" respectively. We called Empirical Relational System a triple: A = (A, • >=, o), where A is a non-empty set of objects, • >= is an empirical relation on A and o is a closed binary (concatenation) operation on A.

Zuse defines a set of properties for measures, which characterise different measurement structures. The most important ones are shown in table 2:

MODIFIED EXTENSIVE STRUCTURE	INDEPENDENCE CONDITIONS	MODIFIED RELATION OF BELIEF
Axiom1: (A, • >=) (weak order) **Axiom2**: A1 o A2 • >= A1 (positivity) **Axiom3**: A1 o (A2 o A3) ≈ (A1 o A2) o A3 (weak associativity) **Axiom4**: A1 o A2 ≈ A2 o A1 (weak commutativity) **Axiom5**: A1 • >= A2 ⇒ A1 o A • >= A2 o A (weak monotonicity) **Axiom6**: If A3 • > A4 then for any A1, A2, then there exists a natural number n, such that A1o nA3 • >A2 o nA4 (Archimedean axiom)	**C1**: A1 ≈ A2 ⇒ A1 o A ≈ A2 o A and A1 ≈ A2 ⇒ A o A1 ≈ A o A2 **C2**: A1 ≈ A2 ⇔ A1 o A ≈ A2 o A and A1 ≈ A2 ⇔A o A1 ≈ A o A2 **C3**: A1 • >= A2 ⇒ A1 o A • >= A2 o A, and A1 • >= A2 ⇒ A o A1 • >= A o A2 **C4**: A1 • >= A2 ⇔ A1 o A • >= A2 o A, and A1 • >= A2 ⇔A o A1 • >= A o A2	**MRB1**: ∀ A, B ∈ ℑ: A • >= B or B • >= A (completeness) **MRB2**: ∀ A, B, C ∈ ℑ: A • >= B and B • >= C ⇒ A • >= C (transitivity) **MRB3**: ∀ A ⊇ B ⇒ A • >= B (dominance axiom) **MRB4**: ∀ (A ⊃ B, A ∩ C = ϕ) ⇒ (A • >= B ⇒ A U C • > B U C) (partial monotonicity) **MRB5**: ∀ A ∈ ℑ: A • >= 0 (positivity)
As we know, binary relation • >= is called weak order if it is transitive and complete: A1 • >= A2, and A2 • >= A3 ⇒ A1 • >= A3 A1 • >= A2 or A2 • >= A1	Where A1 ≈ A2 if and only if A1 • >= A2 and A2 • >= A1, and A1 • > A2 if and only if A1 • >= A2 and not (A2 • >= A1).	

Table 2. Zuse's formal framework properties

There exists five scale types that are defined by admissible transformations. They are, in hierarchical order: nominal, ordinal, interval, ratio and absolute. Each scale type is defined by admissible transformations. Software measurement starts with the ordinal scale ([37]). Measures may be classified in a scale type, depending on whether they assume an extensive structure or not.

When a measure accomplishes this structure, it also accomplishes the independence conditions and can be used on the ratio scale levels.

If a measure does not satisfy the modified extensive structure, the combination rule (that describes the properties of the software measure clearly) will exist or not depending on the independence conditions. When a measure assumes the independence conditions but not the modified extensive structure, the scale type is the ordinal scale.

3.1 Formal verification of the proposed metrics

In relational database systems, and for our purposes, the Empirical Relational System could be defined as: **R** = (R, • >=, o), where R is a non-empty set of relations (tables), • >= is the empirical relation "more or equal complex than" on R and o is a closed binary (concatenation) operation on R. In our case we will choose natural join as the concatenation operation. Depending on the characteristics of the combined tables, natural join can derive in Cartesian product. Furthermore, it is possible to make the natural join through foreign key-primary key or between any columns of two tables defined over the same domain. All these characteristics of the natural join will be useful in order to design the combination rule of the metrics:

1. In order to obtain the combination rule for DRT we may think of several possibilities: the natural join may derive in a Cartesian product or can be made between columns not related by referencial integrity (in both cases the referential paths are not affected by the combination and the final value of the metric is the longest referential path), or the natural join may be made by foreign key (the length of the referential paths may vary, decreasing in one).So, we can generalise and define the combination rule as: DRT(RioRj)=max(DRT(Ri),DRT(Rj)) – v being v a variable.
2. In order to obtain the combination rule for RD, we can observe that if the concatenation (by natural join) between tables is made by foreign key, the number of foreign keys are affected (decreasing in one), and are not affected in other cases. So, we can characterise the combination rule for RD as: RD(RioRj) = RD(Ri) + RD(Rj) –v where v is a variable that can be 0 or 1.

Table 3 presents the results obtained for the metrics with the combination functions described:

Properties	DRT	RD
Axiom 1	YES	YES
Axiom 2	NO	NO
Axiom 3	YES	YES
Axiom 4	YES	YES
Axiom 5	NO	NO
Axiom 6	NO	NO
Ind Cond. 1	NO	NO
Ind Cond. 2	NO	NO
Ind Cond. 3	NO	NO
Ind Cond. 4	NO	NO
MRB 1	YES	YES
MRB 2	YES	YES
MRB 3	YES	YES
MRB4	YES	YES
MRB5	YES	YES

Table 3. Characterization of rule's circumstance measures

So, we can characterise both metrics above the level of the ordinal scale assuming the modified relation of belief.

4. EMPIRICAL VALIDATION

As Basili points out, software engineering is a laboratory science, but there is insufficient analysis and experimentation (Basili, 1998). Typically experimentation is looking for a relationship between two variables, such as process and product characteristics, and may be performed for many purposes: to study process effects, product characteristics, environmental constraints (cost or schedule). Besides, it is necessary to combine experiments to build a body of knowledge that is useful to the discipline.

In this section we summarise an experiment done in order to validate DRT and RD metrics. This empirical validation has been carried out following the experimental method applied to software engineering (Pfleeger, 1995) (Bourque and Côte 1991). Our purpose is to prove that the metrics DRT and RD can be used for measuring the complexity of the relational database schema which influences in its understandability.

4.1 Hypotheses

The formal hypotheses are:

1. Null hypothesis: Different values of metrics do not affect the comprehension of the database schema.
2. Alternative hypothesis 1: The value of the DRT metric affects the comprehension of the database schema.
3. Alternative hypothesis 2: The value of the RD metric affects the comprehension of the database schema.
4. Alternative hypothesis 3: The combination of the DRT and RD metrics affects the comprehension of the database schema.

4.2 Experimental design

Each level of one factor appears with each level of the other one; therefore, we have selected the crossing design. This crossing relationship is defined AxB. For us, A is the RD metric and B is the DRT metric (see Table 4). There were two possible values for RD metric (eight or five) and for DRT metric (two or five). In order to increasing the power of the test, □ has been set to 0.1 instead of the more common level of 0,05 (Briand et al, 1997).

		Factor B (RD)	
		LOW	HIGH
Factor A (DRT)	LOW	2,5	2,8
	HIGH	5,5	5,8

Table 4. Crossed Design for the experiment

4.3 Experimental procedure

The complete documentation was given to each subject and contained all the material related to the four tests: relational schemata, tables with data and the question/answer paper.

Before we started, the following aspects of the experiment were explained, what kind of exercises were to be developed, the material given, how to answer the questions made, and how long each test lasted. When the time of each test finished (ten minutes), the subjects were informed and, immediately, they could change to another test.

4.4 Subjects

The participants in the experiment are Computer Science students at the University of Castilla-La Mancha (Spain), who were enrolled in the databases course lasting two semesters. Until the day of the experiment, the students did not known that they were going to do it. The experiment was developed by 60 students but only 59 were finally accepted.

We have tried to minimise variability among participants by choosing people of the same degree, the third one (the last in Computer Science BSc). Effects of irrelevant variables were minimised making the same tests for all the subjects with the same duration (ten minutes per test).

4.5 Experimental materials

The documentation accompanying each design was consisted of approximately seven pages long and included the schema database, the tables with their rows and the question/answer paper. For each design the database schema had six tables.

The subjects were asked to perform three tasks with the values of the database schema: insert, delete and update. Figure 2 shows the question/answer paper.

Before the subjects took the test, the experiment was conducted on a small set of people.

1. What tables and how many rows in each table are affected if we delete in the Table 5 the row with cod1=210?

Table 1	Table 2	Table 3	Table 4	Table 5	Table 6

2. What tables and how many rows in each table are affected if we update the column X of the row with cod2=11 in the table 3?

Table 1	Table 2	Table 3	Table 4	Table 5	Table 6

3. What tables and how many rows and columns are necessary to add if we want add a new row in the table 4? (Suppose that all the necessary data are news in the database)

Table 1	Table 2	Table 3	Table 4	Table 5	Table 6

Figure 2. Question/answer paper

4.6 Experimental results

There are three major items to consider when choosing the analysis techniques: the nature of the data collected, why the experiment is performed and the type of experimental design used ([29]).

Due to the type of the experiment used, F statistic is the most appropriate technique for obtaining the results (Rohatgi, 1976). Table 5 shows the results for the F-statistic:

Source of Variation	Q_i	Degrees of Freedom	s_i^2	F-Ratio
DRT	18.457	1	18.5	1.67
RD	531.000	1	531	48.1
Interaction	31.339	1	31.3	2.84
Error	2560.304	1	11.0	
Total	3141.102	232		

Table 5. Results of the F-statistic

By comparing these values with $F_{1,232}=2.73$, we can ensure that:

1. Alternative Hypothesis 1. Since $1.67<2.73$, DRT does not affect the results of the experiment. Therefore, alternative hypothesis 1 is invalid because the value of the DRT metric does not affect the results obtained.
2. Alternative Hypothesis 2. Since $48.1>2.73$, RD affects the results of the experiment. As a consequence, alternative hypothesis 2 is valid because the value of the RD metric affects the results obtained.
3. Alternative Hypothesis 3. Since $2.84>2.73$, the interaction of the metrics affects the results of the experiment. As a result, alternative hypothesis 3 is valid because the combination of the values of the DRT and the RD metrics affect the results obtained.

We can conclude by saying that the number of foreign keys in a relational database schema is a more solid indicator of its understandability than the length of the referential paths and that the length of the referential tree is not relevant by itself, but can modulate the effect of the number of foreign keys.

Following (Schneidewind, 1997) RD can be classified as a dominant metric, and DRT is not needed to classify quality (is a redundant metric).

When we obtained the results of the experiment one, some questions arose: Is there another metric useful for measuring of understandability?, can we obtain a more powerful method for measuring understandability maybe with another of our metrics and the RD?. In order to answer these questions, we have designed another experiment, making the RD metric fix and working with another metric D also related to the referential integrity, that we have defined as:

Then, in the second experiment we worked with the D and the RD metrics in order to obtain another relation between these two metrics and the understandability of a relational database schema. All the process developed was similar to the other experiment and the conclusions obtained were that the sum of the square of the number of foreign keys per table in a relational database schema is not an indicator of its understandability and cannot modulate the effect of the number of foreign keys.

5. CONCLUSIONS AND FUTURE WORK

More research is needed into the aspects of software measurement (Neil, 1994) both from a theoretical and from a practical points of view (Glass, 1996). We think it is very interesting to dispose on metrics for relational databases. These metrics can be used to flag outlying schemata for special attention, a strong requirement for low testing and maintenance costs would argue for justify extra managerial attention to a quite significant fraction of the relational database schemata (Chidamber, Darcy and Kemerer, 1998).

We have put forward different measures related with referential integrity in order to measure the complexity of the relational database schemata. These metrics were developed and characterised in accordance with a set of sound measurement principles.

Relational databases measures assume, as object-oriented measures ([37]), more complex properties related to concatenation operation than classic measures. These measures do not assume an extensive structure but can be characterised above the ordinal scale by fulfilling all the properties of modified relations of belief.

We have done some experiments to validate the proposed metrics, but more others are being developed at this moment. However the controlled experiments have problems (such as the large number of variables that cause differences, or the fact that these experiments deal with low level issues, microcosms of reality and small sets of variables) and limits (e.g. they do not scale up, are done in a class in training situations, are made in vitro and face a variety of threats of validity). Therefore, it is convenient to run multiple studies, mixing controlled experiments and case studies (Basili, 1998). For these reasons, a deeper empirical evaluation is under way in collaboration with industrial and public organisations in "real" situations.

We are also adapting our framework to several RDBMS that followed the SQL2 model, simplifying the measures to be taken. In this way, we will be able to give a more precise guide to the metrics usage (Churcher and Shepperd 1995) and develop a tool for automatic metrics collection.

The metrics for relational databases are complemented with others for active databases (Piattini and Díaz, 1999) and object-relational databases (Piattini, Calero, Polo, M. and Ruiz, 1998).

Other interesting future research implies the use of these (and other) metrics for building prediction systems for database projects. In [22] a prediction system based on the number of entities, attributes and relationship of an E/R schema combined with other 4GL-oriented measures (number of menus, edit screens, reports, etc.) is shown.

REFERENCES

Basili, V. R. (1998). Using experiments to build a body of knowledge. Madrid. España.

Batra, D., Hoffer, J.A. and Bostrom, R.P. (1990). A comparison of user performance between the relational and the extended entity relationship models in the discovery phase of database design. CACM, 33 (2): 126-139.

Bourque P., Côte V. (1991). An experiment in Sotware Sizing with structured analysis metrics. J.Systems Software. 15:159-172.

Briand L., Bunse C., Daly J., Differing C., (1997). An experimental comparison of the Maintainability of object-oriented and structured design documents. Proc. Int. Conf. on Software Maintenance, Harrold MJ, Vissagid G (Eds), Bari, 1-3 Oct, 130-138.

Briand, L.C., Differding, C.M. and Rombach, D. (1996). Practical Guidelines for Measurement-Based Process Improvement. Software Process-Improvement and Practice 2, 253-280.

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

Card, D.N. and Glass, R.L. (1990). Measuring Software Design Quality. Englewood Cliffs. USA.

Chidamber, S.R., Darcy, D.P. and Kemerer, C. (1998). Managerial Use of Metrics for Object-Oriented Software: An Exploratory Analysis. IEEE Trans. On Software Engineering, Vol. 24, N° 8, 629-639.

Churcher, N.J. and Shepperd, M.J. (1995). Comments on "A Metrics Suite for Object-Oriented Design". IEEE Trans. on Software Engineering, 21(3): 263-265.
Codd, E.F. (1970). A Relational Model of Data for Larged Shared Data Banks. CACM, 13 (6), 377-387.
Date, C.J. (1995). An Introduction to Database Systems. 6th. Addison-Wesley, Reading, Massachusetts.
Elmasri R, Navathe S. (1997). Database Systems. Second edition. Addison-Wesley. Massachussets
Fenton, N. (1994). Software Measurement: A Necessary Scientific Basis. IEEE Transactions on Software Engineering, 20(3): 199-206.
Fenton, N. and Pfleeger, S. L. (1997). Software Metrics: A Rigorous Approach 2nd. edition. London, Chapman & Hall.
Glass, R. (1996). The Relationship Between Theory and Practice in Software Engineering. IEEE Software, November, 39 (11), 11-13.
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
Henderson-Sellers, B. (1996). Object-oriented Metrics - Measures of complexity. Prentice-Hall, Upper Saddle River, New Jersey.
Jarvenpaa, S. and Machesky, J. (1986). End user learning behavior in data analysis and data modeling tools. Proc. of the 7th Int. Conf. on Information Systems, San Diego: 152-167.
Juhn, S. and Naumann, J. (1985). The effectiveness of data representation characteristics on user validation. Proc. of the 6th Int. Conf. on Information Systems, Indianapolis: 212-226.
Kim, Y.-G. and March, S.T. (1995). Comparing Data Modeling Formalisms. CACM 38(6): 103-115.
Li, H.F. and Chen, W.K. (1987). An empirical study of software metrics. IEEE Trans. on Software Engineering, 13 (6): 679-708.
MacDonell, S.G., Shepperd, M.J. and 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.
McCabe, T.J. (1976). A complexity measure. IEEE Trans. Software Engineering 2(5): 308-320.
Melton, A. (ed.) (1996). Software Measurement. London, International Thomson Computer Press.
Neil, M. (1994) Measurement as an Alternative to Bureaucracy for the Achievement of Software Quality. Software Quality Journal 3 (2), 65-78.
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.
Piattini, M. and Díaz, O. (1999). Metrics for active databases maintainability. Accepted in CAISE'99. Heidelberg, June 16-18.
Pfleeger S.L. (1995). Experimental design and analysis in software engineering. Annals of Software Engineering, 219-253, JC Baltzer AG, Science Publishers.
Pfleeger, S. L. (1997). "Assessing Software Measurement". IEEE Software. March/April, pp. 25-26.
Rohatgi VK (1976). An introduction to probability theory and mathematical statistics. Wiley Series in Probability and mathematical statistics.
Rossi, M. and Brinkkemper, S. (1996). Complexity Metrics for Systems Development Methods and Techniques. Information Systems 21(2): 209-227.
Schneidewind N.F. (1997). Software metrics for quality control. Proceedings of the foiurth

international software metrics symposium. IEEE Computer Society Technical Council on Software Engineering, 127-136

Shoval, P. and Even-Chaime, M. (1987). Database schema design: An experimental comparison between normalization and information analysis. Database 18(3): 30-39.

Sneed, H.M. and 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.

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

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