Software Quality Journal, 9, 79–97, 2001

Table Oriented Metrics for Relational Databases

MARIO PIATTINI, CORAL CALERO AND MARCELA GENERO
{mpiattin, ccalero, mgenero}@inf-cr.uclm.es
ALARCOS Research Group, University of Castilla-La Mancha, Ronda de Calatrava, 5, 13071, Ciudad Real, Spain

Abstract. Developing and selecting high quality software applications are fundamental. It is important that the software applications can be evaluated for every relevant quality characteristic using validated metrics. Software engineers have been putting forward hundreds of quality metrics for software programs, disregarding databases. However, software data aspects are important because the size of data and their system nature contribute to many aspects of a systems quality. In this paper, we proposed some internal metrics to measure relational databases which influence its complexity. Considering the main characteristics of a relational table, we can propose the number of attributes (NA) of a table, the depth of the referential tree (DRT) of a table, and the referential degree (RD) of a table. These measures are characterized using measurement theory, particularly the formal framework proposed by Zuse. As many important issues faced by the software engineering community can only be addressed by experimentation, an experiment has been carried out in order to validate these metrics.

Keywords: databases, quality, metrics, GQM, formal validation, empirical validation

1. Introduction

Nowadays, in a global and increasingly competitive market, quality is a critical success factor for all aspects of economical and organizational success. This quality is particularly important in information systems (IS). Developing and selecting high quality software applications are fundamental. Furthermore, it is important that the software applications can be evaluated for every relevant quality characteristic using validated metrics.

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 are focused on the programme, disregarding data-related quality (Sneed and Foshag, 1998). This neglect could be explained by the fact that until recently, databases have developed in just a secondary role leading to only a minor contribution to the quality of the overall system. Nowadays, databases are used in most of the important IS, becoming their essential core.

In this paper, we propose different metrics to analyze the quality of relational database schemata. Following the ISO/IEC 9126 (ISO, 1994) quality model, several characteristics can be identified in software quality: functionality, reliability, usability, efficiency, maintainability and portability. Taking into account that maintenance accounts for between 60 and 90% of life cycle costs (Card and Glass, 1990; Pigoski, 1997), we have focused our work on maintainability. ISO/IEC 9126 distinguishes five subcharacteristics for maintainability: analyzability, changeability, stability, testability and compliance (see Figure 1). Analyzability, changeability and testability are in turn influenced by complexity (Li and Chen, 1987). However, 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 he considers three components: problem complexity, human cognitive factors and product complexity (Henderson-Selleres, 1996). The last one is our focus.

Therefore, the metrics we define are for measuring the complexity, the internal attribute, of relational databases to assess relational database maintainability, the external attribute.

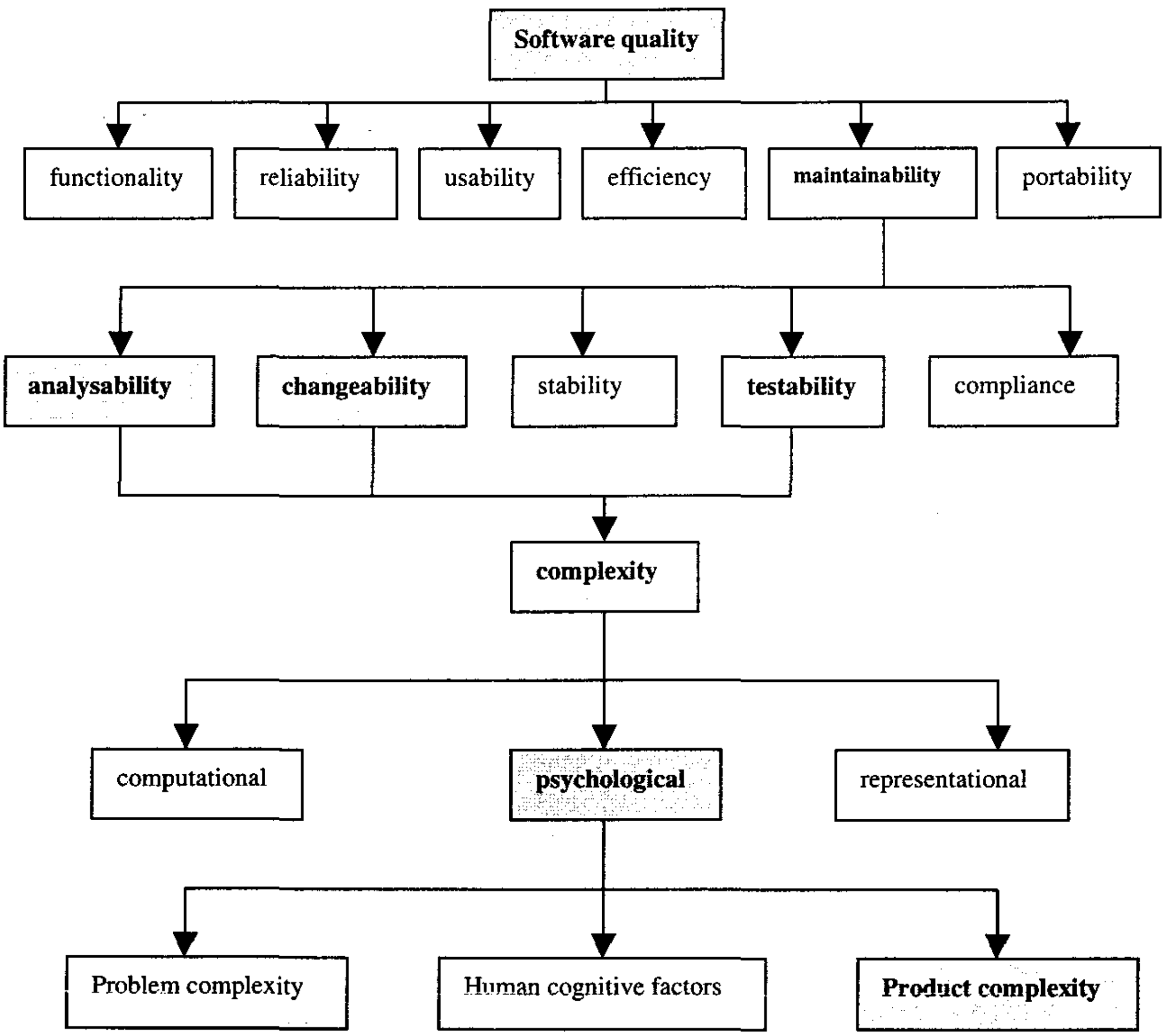


Figure 1. Relationship between products complexity metrics and software quality.

Design product metrics can be sub-divided into intra- and inter-module metrics. Likewise, table complexity can be characterized as *intra-table complexity* where the table in isolation is measured, and *inter-table complexity* where the implicit interaction among tables is measured. Following this distinction, we can define two different kinds of metrics in relational databases: table-oriented (intra-table complexity metrics) and schema-oriented (inter-table complexity metrics), depending on the level on which we are measuring. Schema-oriented metrics are proposed and analyzed in (Calero et al., 2000). In this paper, we propose and validate three table-oriented metrics: number of attributes (NA) of a table, depth of referential tree (DRT) of a table, and referential degree (RD) of a table. These measures are characterized using measurement theory, particularly the formal framework proposed by Zuse (1998).

Section 2 presents the method used for correct metrics definition, the different measures proposed are presented in Section 3. We give a brief introduction to the Zuse's framework in Section 4, using it to characterize the metrics in Section 5. Section 6 presents the experiment performed with the referential integrity related metrics. The conclusions and future work are presented in the last section.

2. A method for metrics definition

Metrics definition must be done in a methodological way, it is necessary to follow a number of steps to ensure the reliability of the proposed metrics. Figure 2 presents ~~the method~~ we applied for the metrics proposal.
In this figure we have three 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

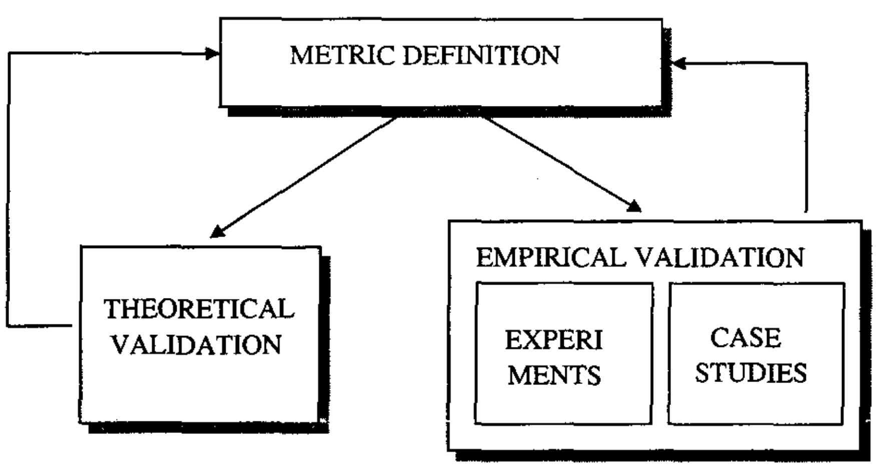


Figure 2. Steps followed in the definition and validation of the metrics.

of these databases. A methodological way to do it is by using the goal-question-metric (GQM) approach (Basili and Weiss, 1984).

- **Theoretical validation.** The second step is the formal validation of the metrics. The formal validation help us to know when and how to apply these metrics.
- **Empirical validation.** The goal of this step is to prove the practical utility of the proposed metrics. There are a lot of ways to prove it but basically we can divide the empirical validation into two: experimentation and case studies. The first one consists in making controlled experiments and the case studies usually work with real data.

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 on the theoretical or the empirical or psychological validations.

3. Measures proposed for relational databases

The relational model proposed by Codd in the late sixties (Codd, 1970), currently dominates the database market. In spite of their diffusion, the only indicator, which has been used to measure the quality of relational database, has been the normalization theory, upon which (Gray et al., 1991) proposed to obtain a normalization ratio.

Normalization alone is not enough to characterize database quality. It is necessary to dispose on metrics specifics for this kind of databases. To obtain them, we can use the GQM approach (Basili and Weiss, 1984), which is based on the fact that any metric can be defined by a top–down schema. The GQM is a three levels model: the conceptual level, where the goals are defined (*goal*), the operational level, where the questions are defined (*question*) and the quantitative level, where the metrics are defined (*metric*).

The application of the GQM approach to relational databases is shown as follows:

Goal. Our goal is to improve the maintainability of the relational databases from the designer point of view.
Question. How the table complexity influences the relational databases maintainability?
Metric. And for answering this question, we proposed the following metrics:

Depth of the referential tree. The DRT of a table A (DRT(A)), is the length of the longest referential path from the table A, counted as the number of arcs on the path. The cycles are only considered once.

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

Number of Attributes. The NA of a table A (NA(A)), is the number of attributes of the table A.

We can apply the previous metrics to the following example (Example 1) taken from Elmasri and Navathe (1999).

```
CREATE TABLE EMPLOYEE
{FNAME          VARCHAR(15)          NOT NULL,
MINIT           CHAR,
LNAME           VARCHAR(15)          NOT NULL,
SSN             CHAR(9)              NOT NULL,
BDATE           DATE,
ADDRESS         VARCHAR(30),
SEX             CHAR,
SALARY          DECIMAL(10,2),
SUPERSSN        CHAR(9),
DNO             INT                  NOT NULL,
PRIMARY KEY (SSN),
FOREIGN KEY (SUPERSSN) REFERENCES EMPLOYEE(SSN),
FOREIGN KEY (DNO) REFERENCES DEPARTMENT(DNUMBER) );

CREATE TABLE DEPARTMENT
( ...
DNUMBER         I NT                 NOT NULL,
MGRSSN        CHA   R(9)             NOT NULL,
...
PRIMARY KEY (DNUMBER)
FOREIGN KEY (MGRSSN) REFERENCES EMPLOYEE(SSN));
```

Example 1. Example of tables definition.

The values for the proposed metrics are presented in Table 1.

Table 1. Values of the metrics in the EMPLOYEE table

	RD	DRT	NA
EMPLOYEE	2	3	10

4. Metrics formal validation

Several frameworks for formal characterization have been proposed. Some of them (Briand et al., 1996; Weyuker, 1988) are based on axiomatic approaches. The goal of this approach is merely definitional by defining formally desirable properties of measures for a given software attribute, so axioms must be used as guidelines for the definition of a measure. Others (Zuse, 1998) are based on measurement theory which specifies the general framework in which the measures should be defined.

Software metrics axioms sets have been developed without a consensus and sometimes without a common understanding of the data to which they will be applied. The main goal of axiomatization 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 a purpose, it cannot be used to validate metrics. We cannot tell whether a measure that does not satisfy the axioms has failed, because either 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 axiomatization 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 outweight the benefits (Kitchenham and Stell, 1997).

The strength of measurement theory is the formulation of empirical conditions from which we can derive hypothesis of reality. In this paper, we will follow the formal framework of Zuse (1998) in order to describe the properties of the metrics defined above. This framework is based on an extension of classical measurement theory, which gives a sound basis for software measures, their validation and the criteria for measurement scales (see the Appendix). As a result of applying this framework we can know to which scale a metric pertains and, behind the scales are hidden the empirical properties of software measures. If we know the scale to which a metric pertains, we have some mathematical information as the operations we can make with that metric or what kind of statistics it is possible to apply to that metric.

In the following section, we adapt this framework to relational databases to verify the fulfilment of the axioms for the metrics proposed in Section 3.

5. Characterization of relational database complexity metrics

In relational database systems, and for our purposes, the empirical relational system could be defined as

$$\mathbf{T} = (T, \bullet >=, \circ)$$

where T is a non-empty set of relations (tables), $\bullet >=$ is the empirical relation "more or equal complex than" on T, while $\circ$ is a closed binary (concatenation) operation on T. In our case we will choose "natural join" as the concatenation operation. Natural join is defined generally as (Elmasri and Navathe (1999)):

$$Q \rightarrow T_{(\langle \text{list1} \rangle *,\ \langle \text{list2} \rangle)} S$$

where $\langle$list1$\rangle$ specifies a list of i attributes of T and $\langle$list2$\rangle$ is a list of i attributes of S. These lists are used in order to make the comparison equality conditions between pairs of attributes. These conditions are afterwards related with the AND operator. Only the list corresponding to the T relation is preserved in Q.

All these characteristics of the natural join will be useful to design the combination rule of the metrics.

5.1. Depth referential tree metric

The DRT measure is a mapping: DRT: $T \rightarrow \Re$ such that the following holds for all tables T_i and $T_j \in T$: $T_i \bullet >= T_j \Leftrightarrow \text{DRT}(T_i) >= DRT(T_j)$.

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 may be created between columns not related by referential 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–primary key link (the length of the referential paths may vary, decreasing in one).
So, we can generalise and define the combination rule as

$$\mathrm{DRT}(T_i \circ T_j) = \max(\mathrm{DRT}(T_i), \mathrm{DRT}(T_j)) - v$$

being v a variable.

5.1.1. DRT as an extensive modified structure

Axiom 1. T1, T2 y T3 being three tables of a relational database schema, it is obvious that

$$\mathrm{DRT(T1)} >= \mathrm{DRT(T2)} \text{ or } \mathrm{DRT(T2)} >= \mathrm{DRT(T1)},$$

and also

$$\text{if } \mathrm{DRT(T1)} >= \mathrm{DRT(T2)} \text{ and } \mathrm{DRT(T2)} >= \mathrm{DRT(T3)}$$
$$\Rightarrow \mathrm{DRT(T1)} >= \mathrm{DRT(T3)}$$

then DRT fulfills the first axiom.

The positivity axiom is not verified by the metrics own definition (when v is distinct of zero). For example, in Figure 3 the value of table T is $\mathrm{DRT}(T) = 3$, however, the value of the table obtained from the concatenation of the T and the T1 tables is $\mathrm{DRT}(T \circ \mathrm{T1}) = 2$.
Associativity and commutativity, axioms three and four, are fulfilled because the natural join operation is both associative and commutative. From Figure 3, it is clear that axiom 5 may not be fulfilled.

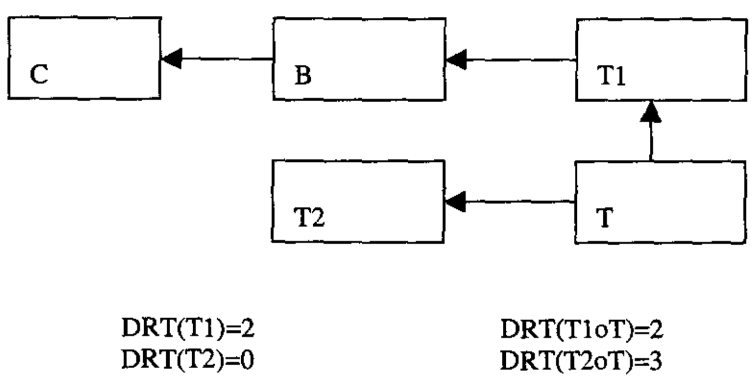


Figure 3. DRT does not fulfil the axiom 5.

Before proving the Archimedean axiom, we verify if the metric is idempotent: it is trivial that if two tables are concatenated (by natural join) more than once, and the referential integrity paths never increase, then the metric is idempotent, and it is therefore possible to ensure that the DRT cannot accomplish the Archimedean axiom. We can conclude that DRT is not an extensive modified structure.

5.1.2. ***DRT and the independence conditions.*** From Figure 4, we can see that the first axiom may not be fulfilled. If DRT does not fulfill C1 or C2. Axioms three and four are not fulfilled because the monotonicity axiom is not fulfilled.

5.1.3. ***DRT and the modified structure of belief.*** Now, we must prove if DRT verifies the modified structure of belief. If the metric meets the weak order, then the first and the second axioms of the modified structure of belief are fulfilled. The third axiom is also fulfilled because if all referential paths of B are included in A, then the value of the longest path of A will be greater than or equal to the value for B.

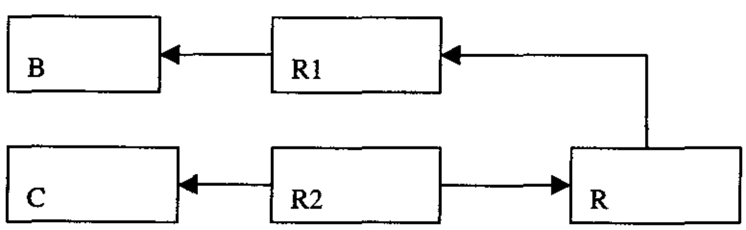
The weak monotonicity axiom is also accomplished because the same referential paths of C are added to A and B. Furthermore, if DRT(A) is greater than DRT(B) it will also be greater after the concatenation of the C paths. The last condition, positivity, is also fulfilled because the length cannot be less than zero. In summary, we can characterize DRT as a measure above the level of the ordinal scale, assuming the modified relation of belief.

5.2. *Referential degree metric*

The RD measure is a mapping: RD: $T \rightarrow \Re$ such that the following holds for all relations T_i and $T_j \in T$: $T_i \bullet >= T_j \Leftrightarrow \mathrm{RD}(T_i) >= \mathrm{RD}(T_j)$.

In order to obtain the combination rule for RD, we must be sure 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 characterize the combination rule for RD as

$$\mathrm{RD}(T_i \circ T_j) = \mathrm{RD}(T_i) + \mathrm{RD}(T_j) - v$$



DRT(R1)=1
DRT(R2)=1

DRT(R1oR)=1
DRT(R2oR)=2

Figure 4. DRT does not fulfil C1.

The formal validation, therefore is analogous to that previously presented. So, in summary, we can characterize RD as a measure above the level of the ordinal scale, assuming the modified relation of belief.

5.3. *Number of attributes metric*

The NA measure is a mapping: NA: $T \rightarrow \Re$ such that the following holds for all relations T_i and $T_j \in T$: $T_i \bullet >= T_j \Leftrightarrow \mathrm{NA}(T_i) >= \mathrm{NA}(T_j)$.

In Figure 5, we can see some of the characteristics that would accomplish the number of attributes when the two tables are combined (by natural join). So, the combination rule for NA can be defined as

$$\mathrm{NA}(T_i \circ T_j) = \mathrm{NA}(T_i) + \mathrm{NA}(T_j) - \mathrm{NA}(T_i \cap T_j)$$

where $\mathrm{NA}(T_i \cap T_j)$ is the number of attributes which are common to (belong to the intersection of) T_i and T_j.

Making the formal verification, NA can be characterized as a measure above the level of the ordinal scale, assuming the modified relation of belief. Table 2 presents the results obtained for the three metrics discussed.

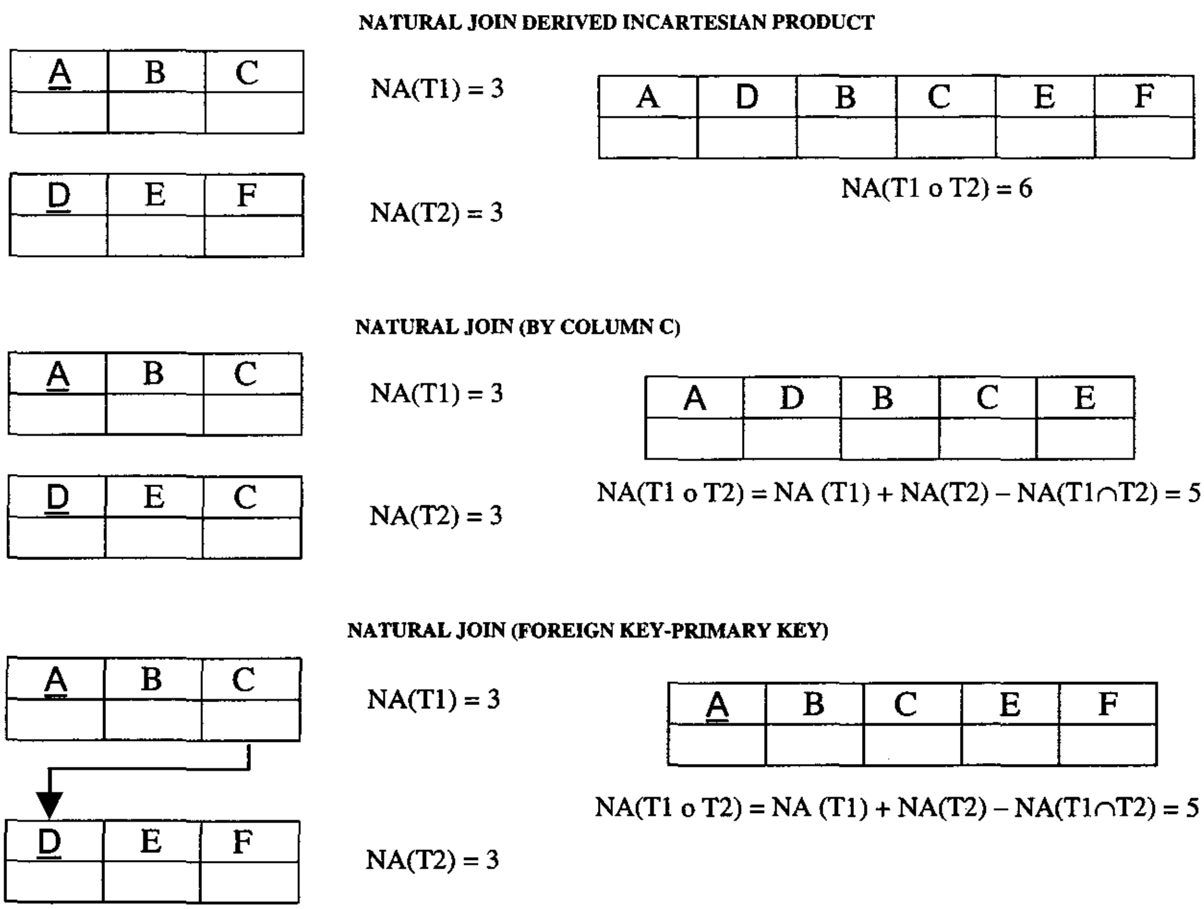


Figure 5. NA of combined tables.

Table 2. Characterization of the table oriented metrics

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

6. Empirical validation

Empirical research can help to characterize the practical utilization of the metrics. The observation of software metrics is necessary in an experimental sense (Brilliant and Knight, 1999) in order to validate them from a practical point of view.

In this section, we resume an experiment carried out 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). We have begun with these two metrics because both are related with referential integrity. Other experiments to empirically validate the NA metric are also being developed.

Our purpose is to prove that metrics DRT and RD can be used for measuring the complexity of a relational database schema. In the experiment we work with the DRT and the RD metrics to determine whether or not some of them are relevant for measuring the complexity of a relational database schema.

6.1. Hypotheses

The formal hypotheses are:

- *Null hypothesis.* Different values of metrics do not affect the analyzability of the database schema.
- *Alternative hypothesis 1.* The value of the DRT metric affects the analyzability of the database schema.
- *Alternative hypothesis 2.* The value of the RD metric affects the analyzability of the database schema.

- *Alternative hypothesis 3.* The combination of the DRT and RD metrics affects the analyzability of the database schema.

6.2. *Subjects*

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

6.3. *Experimental materials*

The documentation accompanying each design was 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 (all the experimental materials can be found in http://alarcos.inf-cr.uclm.es).

The subjects were asked to perform three tasks with the values of the database schema: insert, delete and update. Figure 6 shows the question/answer paper. A

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 6. Question/answer paper.

handout was given to each student, and they had 10 minutes to complete each test. A preliminary experiment was led with a different and smaller set of students to improve the final version.

6.4. *Experimental design*

The experiment attempts to prove whether the DRT and the RD metrics increased the difficulty of understanding the relational schema. We have a factorial model, as all the values a factor can take are combined with all values of the other factor. As we have a constant number of observations in each experimental cell (number of students who answered correctly), the model is balanced. A crossing design as the one described above produces the matrix shown in Table 3 where each value of the matrix is a pair (DRT, RD).

Independent variables. The independent variables are DRT and RD. Each one of these independent variables has two levels, eight or five for RD metric and two or five for DRT metric.

Dependent variables. In our experiments, the variable depends on the number of questions correctly answered in each test. Since what we wanted to measure is understandability, we decided to give our subjects ten minutes per test. This way, all the tests, not necessarily completed, would be taken as valid, and thus our study would focus on the different amounts of correct answers obtained for each test of the experiment.

Controlled variables. We have tried to minimize variability among participants choosing students from the same degree and with the same experience in databases. Effects of irrelevant variables were minimized by making the same trials for all the subjects in the same time (10 minutes).

Procedure. Experiments were made consecutively during an hour session. Before we started, the experiment was explained, what kind of exercises may be developed, the material given, how to respond to the questions, and how much time they had to take each test.

The complete documentation was given to each subject, the documentation contained all the materials related to the four tests: relational schema, tables with data and the question/answer paper. When the time of each test ended, the subjects were informed and, immediately, they could change to another test. The tests were made in different order by the subjects to prevent apprenticeship effects. When the tests

Table 3. Crossed Design for the experiment

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

Table 4. Results of 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		

were marked, the right answers above the total answers were selected for obtaining the results of the experiment.

6.5. *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 (Pfleeger, 1995).

As we have already indicated, we have a factorial model, where we intend to prove whether there is interaction between the two factors. Due to these characteristics, the F statistic is the most appropriate technique to obtain the results (Rohatgi, 1976).

Table 4 shows the results for the F-statistic. In this table, the first column represents the dependent variables, column two the variability (among rows, among columns, of the interaction and for each experimental plot, which is not due neither to the factors we are analysing here nor to their interaction). The third column represents the degrees of freedom and the last one indicates the results obtained for our experiment, these values must be compared to the table values. In each row of the table we have the two factors of the experiment, the interaction, the error and the total.

Power of a statistical test is dependent on three different components: α (error margin), the size of the effect being investigated and the number of subjects. Since the effect size and the number of subjects are constants, increasing α is the only option for increasing the power of the test applied. This way, instead of choosing a value $\alpha = 0.05$ (95% level of confidence) which is more normal, we choose a value $\alpha = 0.1$ (90% level of confidence).

This way, with $\alpha = 0.1$, we look for the statistic value in tables where the numerator is 1 (in the three cases, DRT, RD and interaction, we have a degree of freedom) and the denominator is 232 (degree of freedom of the error), seeing that the value is $F_{1,232} = 2.73$. Comparing the values obtained from the experiment with $F_{1,232}$, we can ensure

- *Alternative hypothesis 1. "The value of the DRT metric affects the analyzability of database schema."* 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.

- *Alternative hypothesis 2. "The value of the RD metric affects the analyzability of database schema."* Since 48.1 > 2.73, RD affects the results of the experiment. Therefore, alternative hypothesis 2 is valid because the value of the RD metric affects the results obtained.
- *Alternative hypothesis 3. "The combination of the DRT and RD metrics values affect the analysability of database schema."* Since 2.84 > 2.73, the interaction of the metrics affects the results of the experiment. Therefore, alternative hypothesis 3 is valid because the combination of the values of the DRT and the RD metrics affects the results obtained.

We can, therefore, conclude that the number of foreign keys in a relational database schema is a solid indicator of its complexity 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.

7. Conclusions and future work

It is important that software products, and obviously databases, are evaluated for all relevant quality characteristics, using validated or widely accepted metrics.

These metrics could help designers, choosing between alternative semantically equivalent schemata, to find the most maintainable one. Due of this, we think it is very important to measure databases and understand their contribution to the overall maintainability of IS.

For having useful metrics we use a method composed by three basic steps: the metrics definition (using the GQM approach), the formal verification (using the Zuse formal framework based on the measurement theory) and the empirical validation (doing a controlled experiment).

We have put forward different measures in order to measure the inter-table complexity that affects the maintainability of the relational database schemata and consequently control its quality.

The results obtained from the formal verification show that relational databases measures assume, as object-oriented measures (Zuse, 1998), more complex properties related to concatenation operation than classic measures. These measures do not assume an extensive structure but can be characterized above the ordinal scale by fulfilling all the properties of the modified relation of belief.

However, in terms of software measurement, much research is needed (Neil, 1994), both from theoretical and from practical points of view (Glass, 1996). So, we have carried out some experiments to validate the proposed metrics and more are being developed at this moment. The presented experiment shows that the number of foreign keys in a relational database schema is a solid indicator of its complexity and that the length of the referential tree is not relevant by itself, but can modulate the effect of the number of foreign keys. However, the controlled experiments have problems (like the large number of variables that causes differences, dealing with low level issues, microcosms of reality and small set of variables) and limits (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 may be more convenient to run multiple studies, mixing controlled experiments with case studies. For these reasons, a more deep empirical evaluation is under way in collaboration with industrial and public organisations in "real-life" situations.

The metrics for relational databases are complemented with others for active (Díaz and Piattini, 1999) and object-relational databases (Piattini et al., 1998). Other interesting future research will be using these (and other) metrics for building prediction systems for database projects. In MacDonell et al. (1997) a prediction system based on the number of entities, attributes and relationships of an E/R schema combined with other 4GL-oriented measures (number of menus, edit screens, reports,...) is shown.

Appendix. Formal framework of Zuse (1998)

Zuse (1998) describes measurement as a detour, "*necessary because humans mostly are not able to make clear and objective decisions or judgments*." Measurement is more than producing numbers, it is the combination of empirical entities with numerical entities. This process starts with the real world, which contains the objects that should be measured. People are interested in the establishment of "empirical relations" between objects, such as "higher than" or "equally high or higher than." These empirical relations will be denoted with the symbols "$\bullet >$" and "$\bullet >=$", respectively. An empirical relational system is a triple

$$A = (A, \bullet >=, \circ)$$

where A is a non-empty set of objects, $\bullet >=$ is an empirical relation on A, and $\circ$ is a closed binary (concatenation) operation on A.

In many cases we are not able to produce directly relevant empirical results due to the difficulty of the question we deal with. There is an "intelligence barrier" that impedes to reduce information without help. With the aid of mathematics and statistics "the intelligence barrier" can be overcome: the empirical objects and relationships are mapped into proper numerical objects and relationships. A numerical relational system can be defined as $B = (\Re, >=, +)$, where $\Re$ are the real numbers, $>=$ a relation on $\Re$, and $+$ a closed binary operation on $\Re$.

A measure is then a mapping u: $A \to \Re$ such that

$$a \bullet >= b \Leftrightarrow u(a) >= u(b); \quad \forall a, b \in A$$

Once the mapping is established, mathematics and statistics can then be used to process the information (e.g., working out means or variances). Measurement theory also leads to conditions where numerical statements can be translated back into

Table AP1. Zuse's formal framework properties

Modified extensive structure	Independence conditions	Modified relation of belief
Axiom1: $(A, \bullet >=)$ (weak order) **Axiom2**: $A1 \circ A2 \bullet >= A1$ (positivity) **Axiom3**: $A1 \circ (A2 \circ A3) = (A1 \circ A2) \circ A3$ (weak associativity) **Axiom4**: $A1 \circ A2 = A2 \circ A1$ (weak commutativity) **Axiom5**: $A1 \bullet >= A2 \Rightarrow A1 \circ A \bullet >= A2 \circ A$ (weak monotonicity) **Axiom6**: If $A3 \bullet > A4$ then for any $A1, A2$, then there exists a natural number n, such that $A1 \circ nA3 \bullet > A2 \circ nA4$ (Archimedean axiom)	**C1**: $A1 = A2 \Rightarrow A1 \circ A = A2 \circ A$ and $A1 = A2 \Rightarrow A \circ A1 = A \circ A2$ **C2**: $A1 = A2 \Leftrightarrow A1 \circ A = A2 \circ A$ and $A1 = A2 \Leftrightarrow A \circ A1 = A \circ A2$ **C3**: $A1 \bullet >= A2 \Rightarrow A1 \circ A \bullet >= A2 \circ A$, and $A1 \bullet >= A2 \Rightarrow A \circ A1 \bullet >= A \circ A2$ **C4**: $A1 \bullet >= A2 \Leftrightarrow A1 \circ A \bullet >= A2 \circ A$, and $A1 \bullet >= A2 \Leftrightarrow A \circ A1 \bullet >= A \circ A2$	**MRB1**: $\forall A, B \in \mathcal{T} : A \bullet >= B$ or $B \bullet >= A$ (completeness) **MRB2**: $\forall A, B, C \in \mathcal{T}$: $A \bullet >= B$ and $B \bullet >= C \Rightarrow A \bullet >= C$ (transitivity) **MRB3**: $\forall A \subseteq B \Rightarrow A \bullet >= B$ (dominance axiom) **MRB4**: $\forall (A \supset B, A \cap C = \phi) \Rightarrow (A \bullet >= B \Rightarrow AUC \bullet > BUC)$ (partial monotonicity) **MRB5**: $\forall A \in \mathcal{T}$: $A \bullet >= 0$ (positivity)
As we know, binary relation $\bullet >=$ is called weak order if it is transitive and complete: $A1 \bullet >= A2$, and $A2 \bullet >= A3 \Rightarrow A1 \bullet >= A3$ $A1 \bullet >= A2$ or $A2 \bullet >= A1$	Where $A1 = A2$ if and only if $A1 \bullet >= A2$ and $A2 \bullet >= A1$, and $A1 \bullet A2$ if and only if $A1 \bullet >= A2$ and not $(A2 \bullet >= A1)$.	

empirical statements. To check whether the measure satisfies the users needs, Zuse proposes an internal validation, based on the comparison between the empirical interpretation of numbers and the empirical statements in the real world.

The combination rule must be defined as

$$u(A1 \circ A2) = f(u(A1), u(A2))$$

where $A1, A2, A1 \circ A2 \in A$ and $f(A1, A2) : A \times A \to A$. This concatenation operation ($\circ$) can be contra-intuitive in the area of software engineering because it is not necessary to combine objects in reality. However, it provides a means for building up complex measurement structures, giving a more precise interpretation of numbers.

On this framework, Zuse defines a set of axioms for measures which gives rise to distinct structures. In Table AP1 we present the most important ones.

In software measurement, there are sufficient frameworks with the next 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. Measures may be classified in a scale type, depending on whether or not they assume an extensive structure. 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 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.

Acknowledgments

The authors thanks the anonymous referees for the careful reading and constructive criticism. 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).

References

Basili, V.R. and Weiss, D.M. 1984. A methodology for collecting valid software engineering data, *IEEE Trans. Software Eng.* SE-10(6).

Bourque P. and Côte V. 1991. An experiment in software sizing with structured analysis metrics, *J. Systems Software* 15: 159–172.

Briand L., Bunse C., Daly J., and Differing C. 1997. An experimental comparison of the maintainability of object-oriented and structured design documents. In Harrold, M.J., and Vissagid G. Eds., *Proc. Int. Conf. on Software Maintenance*, Bari, 1–3 October, pp. 130–138.

Briand, L.C., Morasca, S., and Basili, V. 1996. Property-based software engineering measurement, *IEEE Trans. on Software Eng.* 22(1): 68–85.

Brilliant, S.S. and Kinght, J.C. 1999. Empirical research in software engineering, *ACM Sigsoft* 24(3): 45–52

Calero, C., Piattini, M., Genero, M., Serrano, M., and Caballero, I. 2000. *Metrics for Relational Databases Maintainability*, UKAIS2000, Cardiff, UK, pp. 109–119.

Card, D.N. and Glas, R.L. 1990. *Measuring Software Design Quality*, Englewood Cliffs, Prentice Hall.

Churcher, N.J. and Shepperd, M.J. 1995. Comments on "A metrics suite for object-oriented design," *IEEE Trans. Software Eng.* 21(3): 263–265.

Codd, E.F. 1970. A relational model of data for larged shared data banks, *CACM* 13(6): 377–387.

Díaz, O. and Piattini, M. 1999. *Metrics for Active Databases Maintainability*, CAISE'99. Heidelberg, June 16–18.

Elmasri, R. and Navathe, S. 1999. *Fundamentals of Database Systems*, 3rd ed., Massachussets, Addison-Wesley.

Fenton, N. 1994. Software measurement: A necessary scientific basis, *IEEE Trans. on Software Eng.* 20(3): 199–206.

Fenton, N. and Pfleeger, S.L. 1997. *Software Metrics: A Rigorous Approach*, 2nd ed., 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., and Pengelly. A.D. 1991, Design metrics for database systems, *BT Tech. J.* 9(4): 69–79

Henderson-Sellers, B. 1996. *Object-Oriented Metrics—Measures of Complexity*, Upper Saddle River, NJ, Prentice-Hall.

ISO. 1994. Software product evaluation-quality characteristics and guidelines for their use. *ISO/IEC Standard 9126*, Geneva.

Kitchenham, B. and Stell, J.G. 1997. The danger of using axioms in software metrics, *IEE Proc.-Softw. Eng.* 144(5–6): 279–285.

Li, H.F. and Chen, W.K. 1987. An empirical study of software metrics, *IEEE Trans. on Software Eng.* 13(6): 679–708.

MacDonell, S. G., Shepperd, M. J., and Sallis, P. J. 1997. Metrics for Database Systems: An empirical study. *Proc. Fourth Int. Software Metrics Symp.—Metrics'97*, Albuquerque. *IEEE Computer Society*, pp. 99–107.

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 J.* 3(2): 65–78.

Pfleeger, S.L. 1995. Experimental design and analysis in software engineering, *Ann. Software Eng.*, JC Baltzer AG, Science Publishers, pp. 219–253.

Piattini, M., Calero, C., Polo, M., and Ruiz, F. 1998. Maintainability in object-relational databases, in Van Huysduynen and Peeters (eds.), *Proc. of The European Software Measurement Conf. FESMA 98*, Antwerp, May 6–8, Coombes, pp. 223–230.

Pigoski, T.M. 1997. *Practical Software Maintenance*, New York, Wiley Computer Publishing.

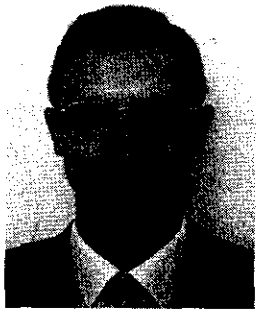
Rohatgi, V.K. (1976). *An Introduction to Probability Theory and Mathematical Statistics*, Wiley Series in Probability and Mathematical Statistics, New York, Wiley.

Schneidewind N.F. 1997. Software metrics for quality control. *Proc. of the Fourth Int. Software Metrics Symp.*, IEEE Computer Society Technical Council on Software Engineering, pp. 127–136.

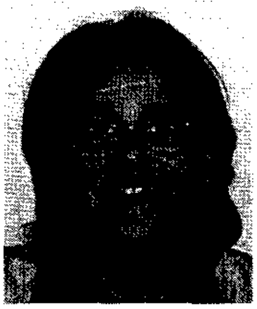
Sneed, H.M. and Foshag, O. 1998. Measuring legacy database structures. *Proc. European Software Measurement Conf. FESMA 98*, Antwerp, May 6–8, Coombes, Van Huysduynen, and Peeters (eds.), pp. 199–211.

Weyuker, E.J. 1988. Evaluating software complexity measures, *IEEE Trans. Software Eng.* 14(9): 1357–1365.

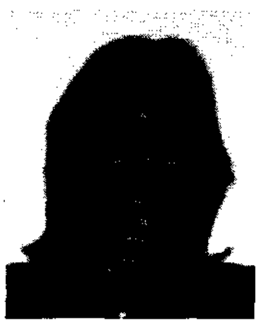
Zuse, H. 1998. *A Framework of Software Measurement*, Berlin, Walter de Gruyter.



Mario Piattini received the MSc and PhD in Computer Science from the Politechnical University of Madrid. Certified Information System Auditor from ISACA (Information System Audit and Control Association). He is an Associate Professor at the Escuela Superior de Informática of the Castilla-La Mancha University. He is the author of several books and papers on databases, software engineering and information systems. He leads the ALARCOS research group of the Department of Computer Science at the University of Castilla-La Mancha, in Ciudad Real, Spain. His research interests are advanced database design, database quality, software metrics, object oriented metrics, software maintenance.



Coral Calero received the PhD in Computer Science, from University of Castilla-La Mancha. She is an Assistant Professor at the Escuela Superior de Informática of the Castilla-La Mancha University in Ciudad Real. She is a member of the Alarcos Research Group, in the same University, specialized in Information Systems, Databases and Software Engineering. She is working on Metrics for Advanced Databases. She is the author of articles and papers in national and international conferences on this subject. She belongs to the ATI association and is a member of its Quality Group.



Marcela Genero is Assistant Professor at the Department of Computer Science in the University of Comahue, in Neuquén, Argentine. She received her MS degree in Computer Science from the National University of South, Argentine in 1989. Actually, she is a PhD student at the University of Castilla-La Mancha, in Ciudad Real, Spain. Her research interests are advanced databases design, software metrics, object oriented metrics, conceptual data models quality, database quality.