# V Jornadas

## Ingeniería de Software
## Bases de Datos

8, 9 y 10 de noviembre
Valladolid

Departamento de Informática
Universidad de Valladolid

EDITORES:
Carlos Delgado
Esperanza Marcos
Jose M. Marqués

ACTAS DE LAS V JORNADAS DE INGENIERÍA DEL SOFTWARE Y BASES DE DATOS (JISBD2000)

EDITORES:
Carlos Delgado
Esperanza Marcos
Jose M. Marqués

ORGANIZADAS POR:
Departamento de Informática
Universidad de Valladolid

ENTIDADES COLABORADORAS:
Universidad de Valladolid. Vicerrectorado de Investigación
Microsoft
Uno-e
Prentice-Hall
Tecsidel
Ayuntamiento de Valladolid



Primera edición, 2000

I.S.B.N.: 84-8448-065-8

Depósito Legal: VA-800/2000

Imprime: Gráficas Andrés Martín S.L.

# COMITÉ DE PROGRAMA DE BASES DE DATOS

## PRESIDENTA

| | |
|---|---|
| Esperanza Marcos | (Universidad Rey Juan Carlos) |

## MIEMBROS

| | |
|---|---|
| José Francisco Aldana | (Universidad de Málaga) |
| Mª José Aramburu | (Universidad Jaume I) |
| Pedro Blesa | (Universidad Politécnica de Valencia) |
| Matilde Celma | (Universidad Politécnica de Valencia) |
| João Correia Lopes | (Universidade do Porto) |
| Dolors Costal | (Universidad Politécnica de Cataluña) |
| Gabriel David | (Universidade do Porto) |
| Oscar Díaz | (Universidad del País Vasco) |
| Jesús García Molina | (Universidad de Murcia) |
| Alfredo Goñi | (Universidad del País Vasco) |
| Arantza Illaramendi | (Universidad del País Vasco) |
| Paloma Martínez | (Universidad Carlos III de Madrid) |
| Eduardo Mena | (Universidad de Zaragoza) |
| Pablo de la Fuente | (Universidad de Valladolid) |
| Joaquim Nunes Aparício | (Universidade Nova de Lisboa) |
| Rui Oliveira | (Universidade do Minho) |
| Mario Piattini | (Universidad Carlos III de Madrid) |
| Antonio Polo | (Universidad de Extremadura) |
| Nieves R. Brisaboa | (Universidad de la Coruña) |
| Pilar Rodriguez | (Universidad Autónoma de Madrid) |
| Félix Saltor | (Universidad Politécnica de Cataluña) |
| José Samos | (Universidad de Granada) |
| Pedro Sousa | (Universidad Técnica de Lisboa) |
| Ernest Teniente | (Universidad Politecnica de Cataluña) |
| Miguel Toro | (Universidad de Sevilla) |
| Toni Urpí | (Universidad Politécnica de Cataluña) |

## COLABORADORES EN EL PROCESO DE REVISIÓN

| | |
|---|---|
| Albert Abello | Rafael Berlanga |
| Coral Calero | Gabriel David |
| Xavier Franch | Francisco José Galan |
| Jorge Enrique Pérez | José Riquelme |
| Antonio Luis Sousa | |

# Comité de Programa de Ingeniería del Software

## Presidente

| | |
|---|---|
| Carlos Delgado Kloos | (Universidad Carlos III de Madrid) |

## Miembros

| | |
|---|---|
| Idoia Alarcón | (Universidad Autónoma de Madrid) |
| Pere Botella | (Universitat Politècnica de Catalunya) |
| Buenaventura Clares | (Universidad de Granada) |
| José Luis Fiadeiro | (Universidade de Lisboa) |
| Xavier Franch | (Universitat Politècnica de Catalunya) |
| Lidia Fuentes | (Universidad de Málaga) |
| Juan Garbajosa | (Universidad Politécnica de Madrid) |
| Juan Hernández | (Universidad de Extremadura) |
| Natalia Juristo | (Universidad Politécnica de Madrid) |
| José Manuel Marqués | (Universidad de Valladolid) |
| Ana Moreira | (Universidade Nova de Lisboa) |
| Juan José Moreno | (Universidad Politécnica de Madrid) |
| José Nuno Fonseca de Oliveira | (Universidade do Minho) |
| João Pascoal Faria | (Universidade do Porto) |
| Oscar Pastor | (Universitat Politècnica de Valencia) |
| Pedro Pastor | (Universidad de Alicante) |
| Mario Piattini | (Universidad de Castilla-La Mancha) |
| Isidro Ramos | (Universitat Politècnica de Valencia) |
| António Rito Silva | (Universidade Técnica de Lisboa) |
| Miguel Toro | (Universidad de Sevilla) |
| Ambrosio Toval | (Universidad de Murcia) |
| José Maria Troya | (Universidad de Málaga) |
| Javier Tuya | (Universidad de Oviedo) |
| Angel Velázquez | (Universidad Rey Juan Carlos) |
| Juan Carlos Yelmo | (Universidad Politécnica de Madrid) |

## Colaboradores en el proceso de revisión

José Carlos Bacelar
José Bernardo Barros
Orlando Belo
José Hilario Canos
Ana Cavalcanti
Dolors Costal Costa
Oscar Dieste
Amador Durán
José Luis Fernández
Xavier Ferré
Marcela Genero Bocco
Rui Gustavo Crespo
Ricardo Imbert
Miguel A Laguna
Esperanza Manso
Manuel Mejías
Begoña Moros
Joaquín Nicolás
Camilo Ocampo Goujon
Macario Polo Usaola
Francisco Ruiz González
José A Troyano
Sira Vegas Hernández



# Comité Organizador

### Presidente

| | |
|---|---|
| José Manuel Marqués | (Universidad de Valladolid) |

### Secretaría

| | |
|---|---|
| Pablo de la Fuente | (Universidad de Valladolid) |

### Miembros

| | |
|---|---|
| Manuel Barrio | (Universidad de Valladolid) |
| Valentín Cardeñoso | (Universidad de Valladolid) |
| Carlos E. Cuesta | (Universidad de Valladolid) |
| Carmen Hernández | (Universidad de Valladolid) |
| Miguel A. Laguna | (Universidad de Valladolid) |
| Esperanza Manso | (Universidad de Valladolid) |
| Mercedes Martínez | (Universidad de Valladolid) |
| Félix Prieto | (Universidad de Valladolid) |
| Belarmino Pulido | (Universidad de Valladolid) |
| Juan J. Rodríguez | (Universidad de Valladolid) |
| Pilar Romay | (Universidad de Valladolid) |
| Jesús Vegas | (Universidad de Valladolid) |
| Diego R. Llanos | (Universidad de Valladolid) |
| Juan Hernández | (Universidad de Extremadura) |
| Mario Piattini | (Universidad de Castilla-La Mancha) |
| Ana Moreira | (Universidade Nova de Lisboa) |

# PRÓLOGO

JISBD2000 reúne dos eventos ya consolidados, las Jornadas de Ingeniería del Software y las Jornadas de Bases de Datos que, en los últimos años, son referencia obligada para todos aquellos investigadores y profesionales interesados en conocer los últimos avances y tendencias en estas áreas. La celebración conjunta de ambos encuentros permitirá afianzar la experiencia iniciada, el pasado año en Cáceres, con tan buenos resultados científicos como organizativos.

En el presente volumen se recogen los trabajos presentados en las V Jornadas de Ingeniería del Software y V Jornadas de Bases de Datos, JISBD2000, celebrados en Valladolid los días 8, 9 y 10 de noviembre de 2000.

JISBD2000 se ha desarrollado en ocho sesiones técnicas, cinco talleres y cinco conferencias invitadas. A estas actividades se ha unido la celebración de las Primeras Jornadas de Bibliotecas Digitales y la Jornada de Seguimiento de proyectos CICYT-TIC-INFO.

Para participar en las sesiones técnicas se recibieron un total de 47 trabajos en Ingeniería del Software y 18 trabajos en Bases de Datos. El esfuerzo de los miembros de los respectivos comités de programa y de colaboradores externos, permitió realizar el proceso de revisión de forma satisfactoria. Nuestro agradecimiento a quienes enviaron sus trabajos y participaron en este proceso.

Cada artículo se revisó por tres evaluadores, quienes tras una ardua labor, dado el número y la calidad de las contribuciones recibidas, seleccionaron 17 trabajos en el área de Ingeniería del Software y 12 en el área de Bases de Datos. Adicionalmente, en el área de Ingeniería del Software, se aceptaron 6 trabajos para su presentación en la modalidad de artículos cortos y 6 en la de poster.

La realización de talleres se incorpora como novedad en esta edición de las JISBD. Su objetivo es proporcionar un foro de en el que un grupo de investigadores y profesionales intercambien opiniones, ideas o resultados sobre temas específicos. Los talleres programados han sido los siguientes: Aplicación de la Ingeniería del Software y las Bases de Datos a los Sistemas de Información Geográfica; Ingeniería del Software Basada en Componentes Distribuidos; Medición, experimentación y calidad en Ingeniería del Software; Bases de Datos Orientadas a Objetos y Lenguajes de especificación en el desarrollo del software.

JISBD2000 es el resultado del trabajo voluntario y desinteresado de muchas personas. Mi agradecimiento a los miembros de los Comités de Programa por su esfuerzo y dedicación, en especial a sus presidentes, Dr. Carlos Delgado y Dra. Esperanza Marcos, a los conferenciantes invitados, a los autores que enviaron sus trabajos y a todas las organizaciones que con su ayuda económica han hecho posible la realización de estas jornadas.

Por último, mi mas sincero reconocimiento a todos los miembros del Comité Organizador, por su entrega y entusiasmo a lo largo de todo el proceso de organización estas jornadas.

*José Manuel Marqués Corral*
Presidente del Comité Organizador

# A Measurement Approach For Conceptual Database Design

Marcela Genero and Mario Piattini

{mgenero, mpiattin }@inf-cr.uclm.es
Grupo ALARCOS
Departamento de Informática
Escuela Superior de Informática
Universidad de Castilla-La Mancha
Ronda de Calatrava, 5 - 13071 - Ciudad Real - ESPAÑA
Tel.: + 34 926 29 53 00 ext. 3715
fax: + 34 926 29 53 54

**Abstract.** Due to the growing demand of quality information systems, continuous attention to and assessment of the quality of information system design, is necessary to produce quality information systems. Databases play an important role in the information system design and justify an independent study of their quality and their contribution to overall information system quality. Conceptual data models lay the foundation of all later design work and also determine what information can be represented in the database. So, conceptual data model quality has a significant impact on the quality of the database which is ultimately implemented, and an even greater impact if we take into account the size and complexity of current databases. In this work, we propose a set of metrics for measuring the complexity of entity relationship diagrams, because in today's database design world it is still the dominant method of conceptual modelling. The availability of metrics allows designers to measure the complexity of entity relationship diagrams in order to improve database quality from the early stages of their life cycle. We put the proposed metrics under theoretical validation following Zuse´s framework, in order to determine the scale at which each of the proposed metrics pertain. And we also carry out an empirical validation in order to ascertain if they may be used as early quality indicators in the information system life cycle.

## 1 Introduction

Due to the growing demand of quality information systems (IS) continuous attention to and assessment of the IS design is necessary to produce quality IS.

The database constitutes only one of the components of an information system. However the central role that the data itself plays in an information system more than justifies an independent study of database design, and their contribution to overall IS quality. For this reason we deal with only these aspects of information system quality that are closely related to databases, specially focusing on their design.

In a typical database design a conceptual data model which specifies the requirements about the database is first built. The conceptual data model lays the


*I.S.B.N.: 84-8448-065-8*

foundation of all later design work and also determines what information can be represented by a database (Feng, 1999). So, its quality has a significant impact on the quality of the database which is ultimately implemented and also on the quality of the application programs that manages its data.

Improving the quality of conceptual data models will therefore be a major step towards the quality improvement of the database development. We will focus on entity relationship (ER) diagrams because in today's database design world it is still the dominant method of conceptual modelling (Muller, 1999).

In practice, evaluation of the quality of conceptual data models takes place in an *ad hoc* manner, if at all. There are no generally accepted guidelines for evaluating the quality of conceptual data models, and little agreement even among experts as to what makes a "good" conceptual data model (Moody and Shanks, 1994).

In general we agree with Krogstie et al. (1995) in the sense that "Most literature provides only bread and butter lists of useful properties without giving a systematic structure for evaluating them". Moreover these lists are mostly unstructured, use imprecise definitions, often overlap, and properties of models are often confused with language method properties (Lindland et al., 1994). In addition to this, these lists are not generally sufficient to ensure quality in practice, because different people will have different interpretations of the same concept. It is necessary to have quantitative and objective measures to reduce subjectivity and bias in the evaluation process.

Recently, some frameworks have been proposed which attempt to address quality in conceptual modelling in a much more systematic way (Lindland et al., 1994; Moody and Shanks, 1994, Moody et al.,1998; Krogstie et al., 1995; Schuette and Rotthowe, 1998), but all of them lack the quantitative assessment of conceptual data model quality.

Metrics can help designers to fix problems, remove non-conforming design attributes, and eliminate unwanted complexity early in the database life cycle. This should reduce the time spent during the different phases of database life cycle, contributing at the same time to a reduction of effort during the development of the application programs that manage its data.

Within the field of software engineering a plethora of metrics has been proposed for measuring software products, processes and resources (Melton, 1996; Fenton and Pflegger, 1997; Henderson-Sellers, 1996). But most of them are focused on programs, disregarding databases and conceptual data models. The only works that propose metrics for conceptual data models are Eick (1991), Gray et al. (1991), Moody (1998) and Kesh (1995).

Although these metric proposals are a good starting point to think about quality in conceptual modelling in a numeric scale, most of them are subjective and lack empirical and theoretical validation.

As in other aspects of Software Engineering, proposing techniques and metrics is not enough, it is also necessary to put them under theoretical and empirical validation, in order to assure their utility in practice. Validation is critical to the success of software measurement (Kitchenham et al., 1995; Fenton and Pflegeer, 1997; Schneidewind, 1992; Basili et al., 1999).

One of the purposes of theoretical validation is the knowledge of the metric scale type (Zuse, 1998). Knowledge of scale type tell us about limitations on the kind of mathematical manipulations that can be performed. The scale type of a measure



affects the types of operations and statistical analyses that can be sensibly applied to the data values (Fenton and Pflegger, 1997).

Equally important is the empirical validation, in order to demonstrate that the proposed metrics really serve in practice for the purpose which they have been created.

In this work we will:

1. propose a set of metrics for ER diagram complexity, taking into account their entities, attributes, and the different kind of relations between entities. (section 2)
2. theoretically validate them following the framework of software measurement proposed by Zuse (1998) with the goal of determining some properties of the proposed metrics, as well as each scale type (section 3).
3. empirically validate the proposed ER complexity metrics with the goal to discover if there exist correlation between them and the time spent through the different phases of the development of the application programs (these programs manage the data conceptually represented in the ER diagrams that the metrics intend to measure) (section 4).

Lastly, in section 5 we will summarise the paper, draw our conclusions, and present our future research directions.

## 2 A proposal of ER diagram complexity metrics

We must be conscious that a general complexity measure is "*the impossible holy grail*" (Fenton, 1994). So in this section we will propose a set of metrics to measure ER diagrams complexity based on their structural complexity (Henderson-Sellers, 1996).

We classify these metrics into the following categories:

### 2.1 Entity metrics

**NE metric.** We define the Number of Entities metric (NE) as the number of entities within the ER diagram.

### 2.2 Attribute metrics

**NA metric.** We define the Number of Attribute metric (NA) as the number of attributes that exist withi the ER diagram, taking into account both entity and relationship attributes. In this number we include simple attributes, composite attributes and also multivalued attributes, each of which take the value 1.

**DA metric.** An ER diagram is minimal when every aspect of the requirements appears once in the diagram, i.e. an ER diagram is minimal if it does not have any redundancies. One of the sources of redundancies in the ER diagrams is the existence of derived attributes. An attribute is derived when its value can be calculated or

deduced from the values of other attributes. We define the Derived Attributes metric (DA) as the number of derived attributes existing in the ER diagram.

**CA metric.** We define the Composite Attributes metric (CA) as the number of composite attributes within an ER diagram. A composite attribute is an attribute composed of a set of simple attributes.

**MVA metric.** The Multivalued Attributes metric (MVA) is defined as the number of multivalued attributes within the ER diagram. A multivalued attribute is an attribute that can take several values for an individual entity.

### 2.2 Relationship metrics

**NR metric.** We define the Number of Relationships metric (NR) as the number of relationships within the ER diagram, taking into account common relationships, represented by the symbol $\diamond$ in the ER diagram.

**M:NR metric.** The M:N Relationships metric (M:NR) is defined as the number of M:N relationships within the ER diagram.

**1:NR metric.** The 1:N Relationships metric (1:NR) is defined as the number of 1:N or 1:1 relationships within the ER diagram..

**N-AryR metric.** The N-ary Relationships metric (N-AryR) is defined as the number of N-ary relationships (not binary) within the ER diagram.

**BinaryR metric.** The Binary Relationships metric (BinaryR) is defined as the number of binary relationships within the ER diagram.

**NIS_AR metric.** We define the Number of IS_A Relationships metric (NIS_AR) as the number of relationships IS_A (generalisation or specialisation) that exist within the ER diagram. In this case, we consider one relationship for each pair child-parent within the IS_A relationship.

**RefR metric.** We define Reflexive Relationships metric (RefR) as the number of reflexive relationships that exist within the ER diagram.

**RR metric.** Another source of redundancy in an ER diagram is the existence of redundant relationships. We define the Redundant Relationship metric (RR) as the number of relationships that are redundant in the ER diagram.

## 3. Theoretical Validation of the Proposed Metrics

In this section we will follow Zuse´s measurement framework (Zuse, 1998) with the goal of determining some properties of the proposed metrics, as well as each scale type. The discussion of scale types is important for statistical operations. Because many empirical and numerical conditions are not covered by a certain scale type, the consideration of the empirical and numerical conditions is necessary and very important, too.



### 3.1 Zuse´s formal framework

This framework is based on an extension of the classical measurement theory. 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 to A and o is a closed binary (concatenation) operation on A. The concatenation operations allow us to define powerful measurement structures (see table 1) which give us a more precise interpretation of numbers. Concatenation operations are directly connected with a measure. But sometimes, a measure assumes a non-intuitive empirical concatenation operation.

Zuse (1998) defines a set of properties for measures, which characterise different measurement structures. The most important ones are shown in table 1.

| MODIFIED EXTENSIVE STRUCTURE | INDEPENDENCE CONDITIONS | MODIFIED RELATION OF BELIEF |
|---|---|---|
| **Axiom1**: (A, • >=) (weak order)<br>**Axiom2**: A1 o A2 • >= A1 (positivity)<br>**Axiom3**: A1 o (A2 o A3) ≈ (A1 o A2) o A3 (weak associativity)<br>**Axiom4**: A1 o A2 ≈ A2 o A1 (weak commutativity)<br>**Axiom5**: A1 • >= A2 ⇒ A1 o A • >= A2 o A (weak monotonicity)<br>**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<br>C2: A1 ≈ A2 ⇔ A1 o A ≈ A2 o A and A1 ≈ A2 ⇔A o A1 ≈ A o A2<br>C3: A1 • >= A2 ⇒ A1 o A • >= A2 o A, and A1 • >= A2 ⇒ A o A1 • >= A o A2<br>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)<br>**MRB2**: ∀ A, B, C ∈ ℑ: A • >= B and B • >= C ⇒ A • >= C (transitivity)<br>**MRB3**: ∀ A ⊇ B ⇒ A • >= B (dominance axiom)<br>**MRB4**: ∀ (A ⊃ B, A ∩ C = ϕ) ⇒ (A • >= B ⇒ A U C • > B U C) (partial monotonicity)<br>**MRB5**: ∀ A ∈ ℑ: A • >= 0 (positivity) |
| As we know, binary relation • >= is called weak order if it is transitive and complete:<br>A1 • >= A2, and A2 • >= A3 ⇒ A1 • >= A3<br>A1 • >= A2 or A2 • >= A1 | Where A1 ≈ A2 if and only if A1 • >= A2 and A2 • >= A1, and A1 • > A2 if and only if<br>A1 • >= A2 and not (A2 • >= A1). | |

**Table 1.** Zuse's formal framework properties

It is important to note that when a metric accomplishes the weak order of the extensive modified structure axiom, it also accomplishes the completeness and the transitivity axioms of the belief structure.

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 (the characterisation of measures above the ordinal scale level is very important because we cannot do very much with ordinal numbers).

In the next subsection we present the formal description of the NA metric. First we define the concatenation operation and the combination function, after we prove the modified extensive structure.



### 3.2 Theoretical validation of the NA metric

For our purposes, the Empirical Relational System could be defined as: E =(E, •>=, o), where E is a non-empty set of ER diagrams, •>= is the empirical relation "equal or more complex than" on E and o is a closed binary (concatenation) operation on E.

In our case we will consider the concatenation operation ERCon. Two ER diagrams, E1 and E2 are concatenated by the concatenation operation ERCon, adding a new relationship between them, as is shown in figure 1.

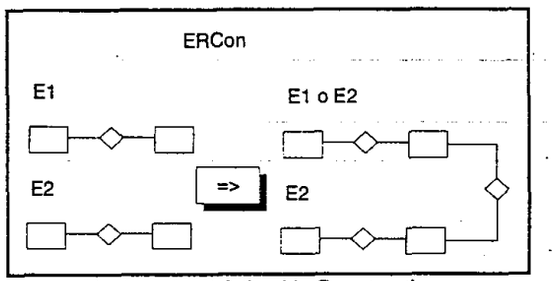


**Fig. 1.** Entity Relationship Concatenation

#### 3.2.1 Theoretical validation of NA metric

NA metric is a mapping: NA:E->ℜ such as the following holds for all ER diagrams Ei and Ej ∈ E: Ei •>= Ej ⇔ NA(Ei) >=NA(Ej)

We can define the combination rule for NA in the following way: NA(Ei o Ej) = NA(Ei) + NA(Ej), ie., the number of attributes of E1oE2, is equal to the sum of the number of attributes of E1 and E2. We do not show attributes in figure 1, for the sake of brevity. We will verify if NA metric fulfils all of the axiom of the Modified Extensive Structure.

Axiom 1. NA fulfils the first axiom of weak order, because if we have two ER diagrams E1 and E2, it is obvious that NA(E1) >= NA(E2) or NA(E2) >= NA(E1) (completeness) and let E1, E2 and E3 three ER diagrams, transitivity is always fulfilled: NA(E1) >= NA(E2) or NA(E2) >= NA(E3), then NA(E1) >= NA(E3).

Axiom 2. NA also fulfils positivity, because the number of attributes of E1 o E2 will be always greater than or equal to the number of attributes of E1. In the case that E2 has no attributes NA(E1 o E2) = NA(E1), and if E2 has attributes NA(E1 o E2) > NA(E1).

Axiom 3. NA also fulfils weak associativity, because the number of attributes does not depend on the order which we associate the ER diagrams to apply the concatenation operation ERCon.

Axiom 4. NA also fulfils weak conmutativity. Taking into account the definition of ERCon, the order in which we concatenate the ER diagrams does not affect the number of attributes.

Axiom 5. NA also fulfils weak monotonicity, because if the number of derived attributes of E1 is greater than or equal to the number of derived attributes of E2, and after we do E1 o E and E2 o E, NA(E1oE)>=NA(E2oE) will result.

Axiom 6. NA also fulfils the Arquimedean axiom. Let E1, E2, E3 and E4 four ER diagrams, and NA(E3)>NA(E4) it is easy to see that one number exits "n" such that NA(E1onE3) > NA(E2onE4), ie. if we concatenate n times E1 with E3, as NA(E3)>NA(E4), for some value of n it will happen that NA(E1 o nE3)>NA(E2 o nE4).

Seeing that NA metric fulfils all of the axiom of the Modified Extensive Structure, we can conclude that this metric is in ratio scale.

Due to the sake of brevity we show the theoretical validation of the rest of metrics in table 2.

| METRICS | COMBINATION RULE | SCALE |
|---|---|---|
| NE | NE(EioEj)=NE(Ei)+NE(Ej) | Ratio |
| DA | DA(EioEj)=DA(Ei)+DA(Ej) | Ratio |
| CA | CA(EioEj)=CA(Ei)+CA(Ej) | Ratio |
| MVA | MVA(EioEj)=MVA(Ei)+MVA(Ej) | Ratio |
| NR | NR(EioEj)=NR(Ei)+NR(Ejj)+1 | Above Ordinal scale |
| M:NR | M:NR(EioEj)=M:NR(Ei)+M:NR(Ej) | Ratio |
| 1:NR | 1:NR(EioEj)=1:NR(Ei)+1:NR(Ej)+1 | Above Ordinal scale |
| N:AryR | N-AryR(EioEj)=N-ryR(Ei)+N:aryR(Ej) | Ratio |
| BinaryR | BinaryR(EioEj)=BinaryR(Ei)+BinaryR(Ej)+1 | Above Ordinal scale |
| NIS_AR | NIS_AR(EioEj)=NIS_AR(Ei)+NIS_AR(Ej) | Ratio |
| RefR | RefR(EioEj)=RefR(Ei)+RefR(Ej) | Ratio |
| RR | RR(EioEj)=RR(Ei)+RR(Ej) | Ratio |

**Table 2.** Theoretical validation of the rest of metrics

## 4. Empirical Validation Of The Proposed Metrics

We want to find out which of the ER diagram complexity metrics defined above are significantly related with the time spent through the different phases of the development of the application programs that manage its data.

In order to perform this validation we have chosen six ER diagrams taken from real implemented information systems. All of them have been built using a tool called Data Architect.

First of all, we briefly describe each ER diagram:

ER 1) WORK_CERTIFICATION: Dedicated to the administration of certificates to the builder CABBSA.

ER 2) ACCOUNTING_ANALYSIS: Dedicated to the administration of accounts for the builder CABBSA.

ER 3) BILLING_ZONE: Dedicated to the control of billing of a series of jobs for the builder CABBSA.

ER 4) SUPPLIERS: dedicated to the administration of CABBSA´suppliers.

ER 5) SOFIA: Dedicated to the control of the offers, evaluations, and product catalogue of one of the units of ERICSSON.



ER 6) WebTime: dedicated to the control of projects and time reporting of Ericsson Business Consulting

Table 3 shows the values of the metrics NE, NA, NR, M:NR, 1:NR and BinaryR and the rest of columns shows the time spent through the application program life cycle: analysis time (A), the design time (D), the implementation time (I), the testing time (T) and the maintenance time (M). All the times are expressed in hours. The column of maintenance time represents the maintenance time in the initial four months from information system delivery. All of the metrics have been collected using a metric tool MANTICA which was developed inside our research group (Calero et al., 1999). We only considered these metrics due to the fact that the rest of the metrics were insignificant, as in each case they took a zero value.

| | NE | NA | NR | M:NR | 1:NR | BinaryR | A | D | I | T | M |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ER 1 | 9 | 98 | 6 | 0 | 6 | 6 | 80 | 40 | 360 | 48 | 14.8 |
| ER 2 | 17 | 72 | 18 | 0 | 18 | 18 | 120 | 40 | 320 | 48 | 22.8 |
| ER 3 | 13 | 84 | 13 | 0 | 13 | 13 | 80 | 40 | 400 | 48 | 14.8 |
| ER 4 | 9 | 80 | 9 | 0 | 9 | 9 | 56 | 24 | 160 | 56 | 14.8 |
| ER 5 | 48 | 178 | 109 | 2 | 101 | 103 | 960 | 480 | 1920 | 160 | 248 |
| ER 6 | 14 | 66 | 19 | 2 | 17 | 19 | 160 | 80 | 640 | 32 | 52 |

**Table 3.** Values of the proposed metrics and the time spent through the application programs life cycle

Pearson´s correlation was used to determine the correlation of the nonparametric data in table 3. The correlation coefficient is a measure of the ability of one variable to predict the value of another variable. Using Pearson´s correlation coefficient, each of the metrics was correlated separately to different times, analysis time, design time, implementation time, testing time and maintenance time.

We wish to test the hypothesis that there is a significant correlation between the current metric data set (NE, NA, NR, M:NR, 1:NR, BinaryR) and the time spent through analysis, design, implementation testing and maintenance phases.

Analysing the Pearson´s correlation coefficients shown in table 4, we can conclude that there exist a high correlation between all of the ER complexity metrics and the time of each phase within the application programs life cycle, as we intuitively think.

| | Analysis time | Design time | Implementation time | Testing time | Maintenance time |
|---|---|---|---|---|---|
| NE | 0.988 | 0.982 | 0.970 | 0.950 | 0.980 |
| NA | 0.940 | 0.944 | 0.911 | 0.974 | 0.922 |
| NR | 0.997 | 0.995 | 0.981 | 0.962 | 0.944 |
| M:NR | 0.696 | 0.705 | 0.772 | 0.505 | 0.742 |
| 1:NR | 0.997 | 0.994 | 0.978 | 0.967 | 0.992 |

| | | | | | |
|---|---|---|---|---|---|
| **BinaryR** | 0.997 | 0.995 | 0.981 | 0.962 | 0.994 |

**Table 4.** Correlation between ER complexity metrics and analysis, design, implementation, testing and maintenance time

Even though the sample size (six real cases) is not enough in order to use this conclusion as a final conclusion, we think that it is a good starting point in order to think about conceptual data models in a numeric terms. We are aware that it is necessary to replicate this experiment with a bigger sample than that which is used in this work. And also it is necessary to perform more experimentation using times directly obtained from the conceptual data models life cycle.

We also have carried out a controlled experiment ascertaining if there exists a significantly correlation between each of the proposed metrics and each of the sub-characteristics that influence the maintainability of a conceptual data model: understandability, legibility, simplicity, analyzability, modifiability, stability, and testability (Genero et al., 2000b). In that experiment, we have analysed the empirical data with a novel data analysis approach based on regression and classification fuzzy trees.

## 5. Conclusions and Future Work

Due to the growing complexity of information systems, continuous attention to and assessment of the conceptual data models is necessary to produce quality information systems. Following this idea, we have presented a set of objective and automatically computed metrics for evaluating ER diagram complexity.

We put them under theoretical validation following Zuse's formal framework in order to demonstrate all of the properties that a metric fulfils and the scale type of each metric. All of the proposed metrics are in ratio scale, which, as was cited above, have an important significance in the scope of software measurement.

We also put them under empirical validation, corroborating that some of the proposed metrics (NE, NA, NR, M:NR, 1:NR, BinaryR) have a high correlation with the time spent through the different phases of the development of the application programs that manages the data conceptually represented in the ER diagrams that the metrics intend to measure.

We are aware of the fact that we must perform more empirical experimentation in order to determine if these metrics could serve as time predictors and could be used as early quality indicators.

Our proposal is a starting point and we require feedback to improve it. However, the absence of other metrics for measuring data structure they serve a purpose in getting database designers to think about the quality of their designs in numeric terms and also in giving managers the possibility of comparing database designs on a numeric scale. As Lord Kelvin put it, one only begins to understand something, when we can measure it (Kelvin, 1954). It is time that this principle be applied to database design as well program design.

We can't disregard the increasing diffusion of the object-oriented paradigm in conceptual modelling. We think that object oriented models are more appropriate than ER diagrams to describe the kind of information systems built nowadays. We are tailoring the proposed metrics (when it is possible) or defining new ones, in order to address the complexity of UML diagrams (Booch, 1998). We have already performed some research related to metrics for measuring class diagrams (Genero et al., 1999; Genero et al., 2000a). As future work we will deal with not only class diagrams but also to use-case diagrams and state diagrams. Furthermore, we will not only address complexity, we also have to focus our research towards measuring other quality factors like the ones proposed in the ISO 9126 (1999).



## Acknowledgements

This research is part of the MANTICA project, partially supported by CICYT and the European Union (CICYT-1FD97-0168).

## References

1. Basili, V., Shull, F. and Lanubile, F. Building knowledge through families of experiments. IEEE Transactions on Software Engineering, Vol. 25 No. 4. (1999) 435-437.
2. Booch, G., Rumbaugh, J. and Jacobson, I. The Unified Modeling Language User Guide. Addison-Wesley, (1998)
3. Calero, C., Pascual, C., Serrano, M. A. and Piattini, M. Measuring Oracle Database Schema. Computers and Computational Engineering in Control, (Cap. 42), World Scientific Engineering Society, (1999) 237-243
4. Eick, C. A Methodology for the Design and Transformation of Conceptual Schemas. Proc. of the 17th International Conference on Very Large Data Bases. Barcelona (1991)
5. Feng, J. The "Information Content" problem of a conceptual data diagram and a possible solution. Proceedings of the 4th UKAIS Conference: Information Systems-The Next Generation, University of York, (1999) 257-266
6. Fenton, N. Software Measurement: A Necessary Scientific Basis. IEEE Transactions on Software Engineering, Vol. 20 No. 3. (1994) 199-206
7. Fenton, N. and Pfleeger, S. L. Software Metrics: A Rigorous Approach. 2nd. edition. London, Chapman & Hall, (1997)
8. Genero, M., Manso, Mª E., Piattini, M. and García, F. Assessing the Quality and the Complexity of OMT Models. 2nd European Software Measurement Conference - FESMA 99, Amsterdam, The Netherlands, (1999) 99-109
9. Genero, M., Piattini, M. and Calero, C. Métricas para jerarquías de agregación en diagramas de clases UML. Memorias de las Jornadas Iberoamericanas de Ingeniería de Requisitos y ambientes de Software, IDEAS'2000, Cancún, México, (2000a) 373-384
10. Genero, M., Jiménez, L. and Piattini, M. Measuring the Quality of Entity Relationship Diagrams. Entity Relationship 2000, Salt Lake City, USA, (2000b)
11. Gray, R., Carey, B., McGlynn, N. and Pengelly A. Design metrics for database systems. BT Technology, Vol. 9 No. 4. (1991)