# The 3 rd. European Software Measurement Conference

# FESMA - AEMES 2000

Letter of Presentation

Committee Members

Explicative Document

Sessions

Wednesday 18 oct.

Thursday 19 oct.

Functional Size Measurement Methods - COSMIC-FFP: Design and Field Trials (Word)

An Approach to Evaluate the Complexity of Conceptual Database Models (Word)
An Approach to Evaluate the Complexity of Conceptual Database Models (PowerPoint)

Functional and Technical Software Measurement: Conflict or Integration? (Word)

Investigating the Applicability of Traditional Test Adequacy Criteria for Object-Oriented Programs (Word)

Function Point Approximation with the Five Logical Components (Word)

Object-Oriented Metrics - A Survey (Word)
Object-Oriented Metrics - A Survey (PowerPoint)

Exploring Cost Estimation Models in the Context of Continuing Software Evolution (Word)
Exploring Cost Estimation Models in the Context of Continuing Software Evolution (PowerPoint)

Guidelines for Evaluation and Improvement of Reuse and Experience Repository Systems Through Measurement Programs (Acrobat Reader)

Object Oriented Software Metrics and the Testing Activity: Some Empirical Results (Word)

Software Project Estimation Based on Knowledge Acquisition (Word)
Software Project Estimation Based on Knowledge Acquisition (PowerPoint)

Software Product Measurement for Supplier Evaluation (Word)
Software Product Measurement for Supplier Evaluation (PowerPoint)

Friday 20 oct.

Click on the links to see the related documents. If your browser has the correct plug-ins for this kind of documents, you will be able to see them directly in your browser. If not, you can also see the documents using applications such as Word, PowerPoint and Acrobat Reader. When you click on the links, should you not be able to open the documents (in case you do not have the plug-ins, or the applications above mentioned), please click here to instal the viewers included in the CD-ROM.

# The 3 rd. European Software Measurement Conference

# FESMA - AEMES 2000

Letter of Presentation

Committee Members

Explicative Document

Sessions

Click on the links to see the related documents. If your browser has the correct plug-ins for this kind of documents, you will be able to see them directly in your browser. If not, you can also see the documents using applications such as Word, PowerPoint and Acrobat Reader. When you click on the links, should you not be able to open the documents (in case you do not have the plug-ins, or the applications above mentioned), please click here to instal the viewers included in the CD-ROM.

asociación española de métricas del software

Madrid, 9th October 2000

Dear Colleague,

We are honored to welcome you to *The 3rd. European Software Measurement Conference,* organized by FESMA- AEMES. As you already know, the Congress will be held at the Hotel Meliá Castilla from Wednesday 18th to Friday 20th October.

We believe that the 3rd Congress edition will be the melting pot of the expertise on the software matter, metrics procedures and also a great opportunity to meet colleagues from all over the world willing to achieve and to develop all the newest tendencies and advances towards the software metric world.

During these three days, the Congress Secretariat and its staff will be at your entire disposition to make your stay in this congress, as comfortable and enjoyable as possible.

We hope that you will find all the lectures and presentations interesting. Please do not hesitate to contact us for further information.

With our best wishes,

José Carrillo

Geert Poels
Co-Presidents of the Congress

## COMMITTEE MEMBERS OF THE THIRD EUROPEAN SOFTWARE MEASUREMENTS CONFERENCE

**Honorific President**
**Presidente de Honor**

Regional Minister of Treasury and Finance, Mr. Manuel Cobo Vega
Excmo. Sr. Consejero de Presidencia de Hacienda
de la Comunidad Autónoma de Madrid

**Honorific Committee**
**Comité de Honor**

**Polytechnic University of Madrid (Spain)**
Rector Magnífico de la Universidad Politécnica de Madrid
Excmo. Sr. D. Saturnino de la Plaza Pérez

**ESI, Managing director**
Director general del ESI
Mr. Manuel Prego

**FESMA, Managing Director**
Director Ejecutivo de FESMA
Mr. Martin Hooft

**Madrid's regional Computing Service, Informática de la**
Comunidad Autónoma de Madrid (ICM)
Mr. Ricardo Domingo Bautista

**Arthur Andersen's Partner**
Socio de Arthur Andersen
Mr. Juan Miguel Ramos

**Compass Analysis, S.A., Deputy Manager**
Director delegado Compass Analysis, S.A.,
Mr. Fernando Marcos Pascual,

**Software AG, The XML Company,**
Director general de Software AG, The XML Company,
Mr. José Ignacio Millán Valderrama

**Congress Presidency**
**Presidencia del Congreso**

**Polytechnic University of Madrid (Spain)**
Universidad Politécnica de Madrid
Mr. José Carrillo,

**Catholic University of Lovaine (Belgium)**
Universidad Católica de Lovaina
Mr. Geert Poels,

**Secretariat of Organization and Financing**
**Secretaría de Organización y Finanzas**
FESMA Managing Director, Mr. Martin Hooft

**Technical Committee, Committee of the European Federations**
**Comité Técnico de las Federaciones Europeas**

**DANMET, Danish Software Metrics Association**
Mr. Erik Hansen

**DASMA, Deutsche Anwendergruppe fuer Software Metrik and Aufwandschaetzung**
Mr. Manfred Bundschuhm

**FISMA, Finnish Software Metrics Association**
Mr. Risto Nevalainen

**FFPUG, French Function Point User Group**

**FPUGA, Function Point Users Group Austria**
Mr. Gerald Rehling

**TI / KVIV, Belgisch Genootschap voor Software Metrics Belgium**
Mr. Bruno Peeters

**NESMA, Netherlands Sofware Metrics Association**
Mr. Ernst van Waning

**UKSMA, United Kingdom Software Metrics Association**
Mr. Harry Coombes
**AEMES, Asociación Española de Métricas del Software**
Mr. José Carrillo

**AEMES Board of Directors**

# AN APPROACH TO EVALUATE THE COMPLEXITY OF CONCEPTUAL DATABASE MODELS

Marcela Genero, Mario Piattini, Coral Calero
Grupo ALARCOS
University of Castilla-La Mancha
Ronda de Calatrava, 5
13071, Ciudad Real (Spain)
e-mail: {mgenero, mpiattin, ccalero}@inf-cr.uclm.es

## ABSTRACT

Nowadays, databases are the core of information systems and their quality have a significant impact on the quality of the information system which is ultimately implemented. Database quality deals with different aspects: DBMS quality, data model quality (both conceptual and logical) and data quality. We only refer in this work to data model quality, specially focused on conceptual data models. We focus on entity relationship diagrams because in today's database design world it is still the dominant method of conceptual modelling. The goal of this work is to propose a set of metrics for measuring entity relationships diagrams complexity, allowing thus database designers to measure the complexity of their designs since the early stages of information systems life-cycle. In order to empirically validate the proposed metrics, we carried out a case study with the goal of discovering if there exist correlation between them an the maintenance time of the application programs that manage the data conceptually represented in the ER diagrams that the metrics intend to measure.

**Keywords:** Conceptual Data Model Metrics, Complexity Metrics, Empirical Validation, Database Quality

## 1. INTRODUCTON

The great demand of quality information system has made quality more of a discriminator between products than ever before, becoming a factor for success.

Nowadays, databases are the core of information systems (IS) and their quality have a significant impact on the quality of the IS which is ultimately implemented.

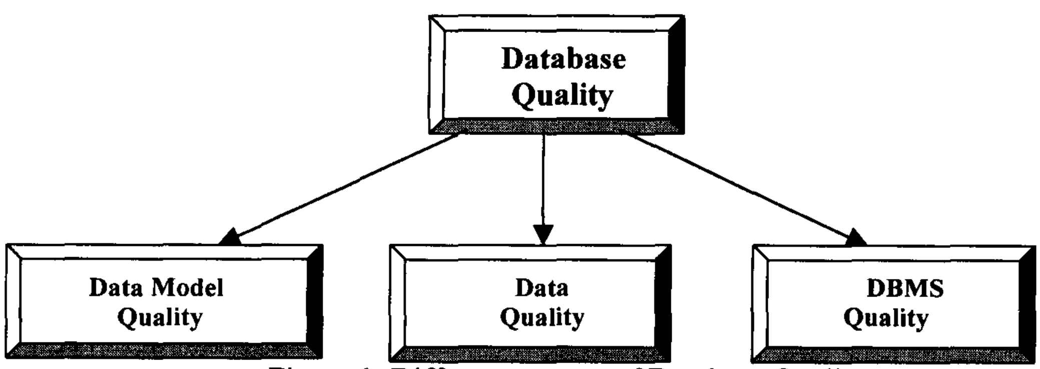


Figure 1. Different aspects of Database Quality

Database quality, as is shown in figure 1, deals with different aspects: DBMS quality, data model quality (both conceptual and logical) and data quality. We only refer in this work to data model quality, specially focused on conceptual data models. In our research group, other works have been presented focused on quality in logical data models, such as relational, and object relational ones (Calero et al., 1999; Calero et al., 2000), and also to data quality (Piattini, et al., 2000). In order to assess DBMS quality we can use an international standard like ISO/IEC 9126 (ISO, 1999), or some of the existing product comparative studies (see, for example, (Barry, 1996) for ODBMS evaluation).

We are aware that 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 the aspects of information system quality that are closely of databases, specially focusing on their conceptual design.

The conceptual data model which specifies the requirements about the database is the first product within the database life cycle. Moreover conceptual data models determine what information can be represented by an information system (Feng, 1999). So that, it is really important to build them as "good" as possible because the cost of correcting a mistake introduced in the first stages of software development grows up exponentially with the project advance (Boehm, 1981). In practice, evaluation of the quality of conceptual data models takes place in an *ad hoc* manner, if at all. Very few organisations have a formally defined process for assuring the quality of conceptual data models as part of IS development.

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". Even more 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). Moreover, 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.

Software engineers have proposed a plethora of metrics for software products, processes and resources (Melton, 1996; Fenton and Pflegger, 1997). Many of the metrics and quality models currently available can be applied only after a product is complete, or nearly complete. They rely upon information extracted from the implementation of the product. This provide information too late to help improve internal product characteristics prior the completion of the product.

Some of the works that propose metrics for conceptual data models are Eick (1991) Gray et al. (1991), Moody (1998) and Kesh (1995). Although all of 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. Thus, there is a need for metrics and models that can be applied in the early stages of development, particularly in what we concern applied to conceptual data models, to ensure that design have favourable internal properties that will lead to the development of quality IS. We will focus on entity relationship (ER) diagrams (Chen 1976; Teorey,

1999) because in today's database design world it is still the dominant method of conceptual modelling (Muller, 1999).

Conceptual data model quality depends on several factors: functionality, reliability, usability, efficiency, maintainability and portability (ISO, 1999). Our focus is on maintainability because maintenance arrange between 60 and 90 percent of life cycle cost and it is considered the most important concern for modern IS department (Frazer,1992; McClure, 1992; Pigoski, 1997). The International Standard ISO/IEC 9126 distinguishes five sub-characteristics for maintainability: analysability, changeability, stability, testability and compliance. Most of these maintainability sub-characteristics are in turn influenced by complexity (Li and Chen, 1987).

In this work we propose a set of metrics for ER diagrams complexity, allowing thus database designers to measure the complexity of their designs since the early stages of information systems life-cycle. This measurement approach would give designers an opportunity to fix problems, remove non-conforming design attributes, and eliminate unwanted complexity early in the development cycle. This should reduce rework during implementation and maintenance.

This work is organised thus:

- In section 2 we will define a framework for developing and validating metrics.
- In section 3 we will propose a set of metrics for measuring the structural complexity ER diagram complexity, taking into account their entities, attributes, and the different kind of relations between entities.
- In section 4 we will put the proposed metrics under empirical validation the goal to discover if there exist correlation between them an the maintenance time of the application programs that manage the data conceptually represented in the ER diagrams that the metrics intend to measure.
- Lastly, in section 5 we will summarise the chapter, draw our conclusions, and present our future work in the field of data models metrics for assuring quality in conceptual modelling.

**Keywords:** Conceptual Data Models Metrics, Complexity Metrics, Empirical Validation, Database Quality

## 2. A FRAMEWORK FOR DEVELOPING AND VALIDATING METRICS

As we have said previously, our goal is to define metrics for ER structural complexity. But metrics definition must be done in a methodological way, it is necessary to follow a number of steps for ensure the reliability of the proposed metrics. Figure 2 presents the method we apply for developing and validating metrics.

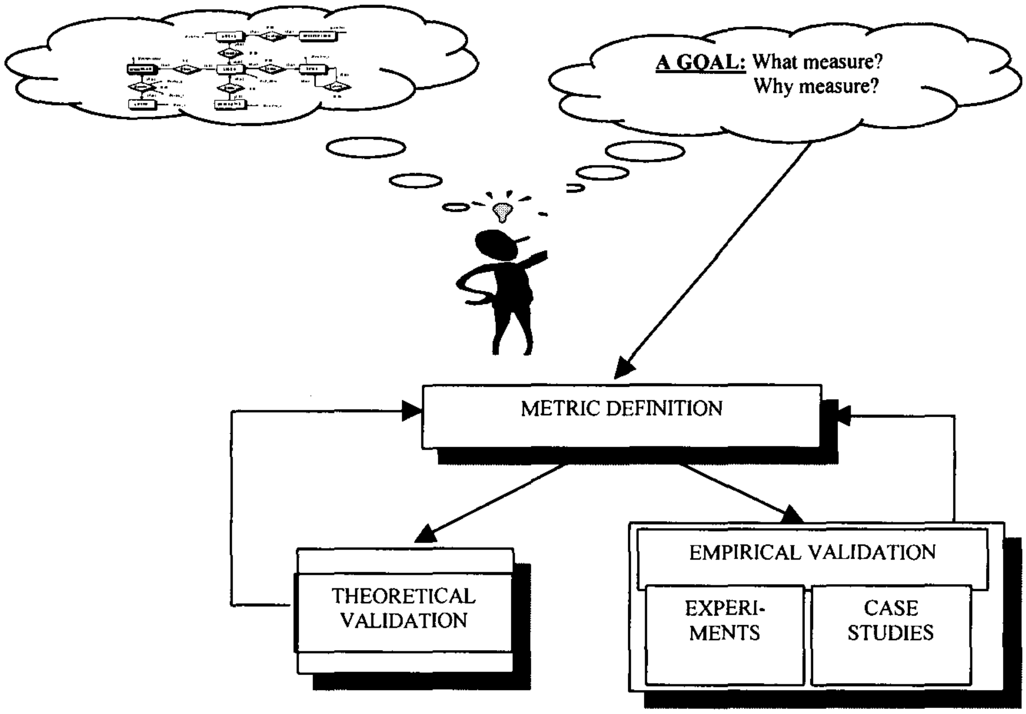


Figure 2. Steps followed in the definition and validation of metrics

After deciding what quality characteristic you want to measure and why, we follow the following steps:

1. **Metric 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 ER diagrams we want to measure. Also metrics must correspond to the specific goals of measurement.

2. **Theoretical validation.** The second step is the theoretical validation of the metrics. While in the software measurement arena there is uniform agreement that we need to put metrics under theoretical validation there is little agreement what is the best way to do that. Several attempts have been made to list a set of rules for theoretical validation (Weyuker, 1988; Kitchenham et al., 1995; Schneidewind, 1992; Briand et al., 1996). However most of these frameworks have been criticised and there is not yet a standard, accepted way of theoretically validate a measure. As Van den Berg and Van den Broek (1996) said a standard on theoretical validation issues in software measurement is urgently required.
   There are two main tendencies in metric theoretical validation: the frameworks based on axiomatic approaches and the ones based on the measurement theory. The goal of the first ones is merely definitional by defining formally desirable properties for measures for a given software attribute, so axioms must be used as guidelines for the definition of a measure. The most well-known frameworks of this type are those proposed by Weyuker (1988), Briand et al. (1996) and Morasca and Briand (1997). Software metrics axioms sets have been developed without a consensus and sometimes without a common understanding of the data to which they apply. The main goal of axiomatisation in software metrics research is the clarification of concepts to ensure that new metrics are in some sense valid. The measurement

theory based frameworks (such as Zuse 1998 or Withmire, 1997) specify a general framework in which measures should be defined. The strength of measurement theory is the formulation of empirical conditions from which we can derive hypothesis of reality.

3. **Empirical validation.** The goal of this step is to prove the practical utility of the proposed metrics. There are a lot of ways to made it but basically we can divide the empirical validation in two: experimentation and case studies. The experimentation is usually made using controlled experiments and the case studies usually work with real data. Both of them are necessary, the controlled experiments for having a first approach and the case studies for making the results stronger. Also is necessary the experiment replication because the isolate results of an experiment it is difficult to understand how widely applicable the results are and, thus, to assess the true contribution to the field (Basili et al., 1999).

As steps 2 and 3 are independent they can be done simultaneously. This process is iterative, and each step can feed the others, which lead us to accept, improve, change or discard the proposed metrics.

## 3. A PROPOSAL OF ER DIAGRAM COMPLEXITY METRICS

As complexity is a multidimensional attributes it is not advisable to combine different aspects of this attribute into a single measurement unless you have a model or theory to support you (Fenton, 1994). Henderson-Sellers (1996) distinguishes three types of complexity, among which he quoted "product or structural complexity", which is our focus when we refer to the concept of complexity. 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:

### 3.1 Entity Metrics

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

### 3.2 Attribute Metrics

- **NA metric.** We define the Number of Attribute metric (NA) as the number of attributes that exist within 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.

### 3.3 Relationship Metrics

- **NR metric**. We define the Number of Relationships metric (NR) as the number of relationships within 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.

- **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.

A theoretical validation of some of the proposed metrics can be found in (Genero et al., 2000a; Genero et al., 2000b). The empirical validation is explained in the next section.

## 4. EMPIRICAL VALIDATION OF THE PROPOSED METRICS

Defining metrics is a very hard task, because sometimes we define metrics with the intention of measuring something but when we put them in practice, we realise that they do not work as we have expected. So, it is essential to put metrics under empirical validation.

In order to validate the metrics proposed in section 2 we have carried out a case study. We have chosen five ER diagrams taken from real implemented information systems. All of them have been built using a tool called Data Architect.

Table 1 shows the values of the metrics NE, NA, NR, M:NR, 1:NR and BinaryR and the last column shows maintenance time (expressed in hours) in the initial six months from system delivery. All of the metrics have been collected using a metric tool MANTICA which was developed inside our research group. 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 | Maintenance time (hours) |
|---|---|---|---|---|---|---|---|
| ER 1 | 9 | 98 | 6 | 0 | 6 | 6 | 12 |
| ER 2 | 17 | 72 | 18 | 0 | 18 | 18 | 17 |
| ER 3 | 13 | 84 | 13 | 0 | 13 | 13 | 12 |
| ER 4 | 9 | 80 | 9 | 0 | 9 | 9 | 12 |
| ER 5 | 48 | 178 | 109 | 2 | 101 | 103 | 208 |

Table 1. Values of the proposed metrics and the maintenance time

Pearson´s correlation was used to determine the correlation of the nonparametric data in table 1. 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 the maintenance time.

We would like 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 maintenance time of the application programs that manage the data conceptually represented in the ER diagrams that the metrics intend to measure.

Analysing the Pearson´s correlation coefficients shown in table 2, we can conclude that a high correlation exists between all of the metrics and the maintenance time, as we intuitively think.

| Mainte-nance Time With NE Metric | Mainte-nance Time with NA Metric | Mainte-nance Time with NR Metric | Mainte-nance Time with M:NR Metric | Mainte-nance Time With 1:NR Metric | Mainte-nance Time with BinaryR Metric |
|---|---|---|---|---|---|
| 0.989 | 0.971 | 0.997 | 1 | 0.996 | 0.996 |

Table 2. Correlation between ER complexity metrics and the maintenance time

Even though the sample size (five 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 study with a bigger sample than that which is used in this work.

## 5. CONCLUSIONS AND FUTURE WORK

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

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 maintenance time 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 experimentation in order to validate these metrics as maintenance time predictors. Specially, we need more real data related to ER diagrams maintenance time, instead of data related to application programs.

We also have carried out a controlled experiment in order to build a maintainability prediction model from the proposed each of the proposed metrics (Genero et al., 2000d). This prediction model is built using a novel method for induction of fuzzy rule systems.

We want to highlight that our proposal cannot be considered as a final proposal. Instead, it is a starting point and we require feedback to improve it.

Due to the increasing and fast diffusion of the object oriented (OO) paradigm, we are tailoring the proposed metrics (when it is possible) or defining new ones, in order to address the complexity of IS using UML (Booch, 1998). We have already performed some research regarding OO conceptual modelling (Genero et al., 1999; Genero et al., 2000c). 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).

Now we are working on building a tool to analyse measurement empirical data, using a novel data analysis approach based on regression and classification fuzzy trees (Genero, et al., 2000d).

## ACKNOWLEDGEMENTS

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

## REFERENCES

Barry, D., *The Object Database Handbook*, John Wiley & Sons, New York, 1996.

Basili, V., Shull, F. and Lanubile, F. (1999). Building knowledge through families of experiments. *IEEE Transactions on Software Engineering*, 25(4), 435-437.

Boehm, B.W. (1981). *Software Engineering Economics*. Prentice-Hall, USA

Booch, G., Rumbaugh, J. and Jacobson, I. (1998). *The Unified Modeling Language User Guide*. Addison-Wesley.

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

Calero, C., Piattini, M., Ruiz, F. and Polo, M. (1999) Validation of metrics for Object-Relational Databases, *International Workshop on Quantitative Approaches in Object-Oriented Software Engineering (ECOOP99),* Lisbon (Portugal), 14-18 June

Calero, C., Piattini, M., Genero, M., Serrano, M. and Caballero, I. (2000). Metrics for Relational Databases Maintainability, *UKAIS2000*, Cardiff, UK.

Chen, P. (1976). The Entity-Relationship Model: toward a unified view of data. *ACM Transactions on Database Systems*, 1, 9-36.

Eick, C (1991). A Methodology for the Design and Transformation of Conceptual Schemas. *Proc. of the 17$^{th}$ International Conference on Very Large Data Bases.* Barcelona.

Feng, J. (1999). The "Information Content" problem of a conceptual data diagram and a possible solution. *Proceedings of the 4$^{th}$ UKAIS Conference: Information Systems-The Next Generation*, University of York, 257-266.

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.* 2$^{nd}$. edition. London, Chapman & Hall.

Frazer, A. (1992). Reverse engineering-hype, hope or here?. In: P.A.V. Hall, *Software Reuse and Reverse Engineering in Practice*. Chapman & Hall.

Genero, M., Manso, Mª E., Piattini,M., García, F. (1999). Assessing the Quality and the Complexity of OMT Models. *2$^{nd}$ European Software Measurement Conference - FESMA 99*, Amsterdam, The Netherlands, 99-109.

Genero, M., Piattini, M. and Calero, C. (2000a). Formalization of Metrics for Conceptual Data Models. *UKAIS 2000*. Cardiff, 26-28 April.

Genero, M., Piattini, M., Calero, C. Serrano, M. (2000b). Measures to get better quality databases. *ICEIS 2000*. Stafford, 4-7 July.

Genero, M., Piattini, M. and Calero, C. (2000c). Métricas para jerarquías de agregación en diagramas de clases UML. *Memorias del Jornadas Iberoamericanas de Ingeniería de Requisitos y ambientes de Software, IDEAS'2000*, Cancún, México, 5-7 Abril, 373-384.

Genero, M., Jimémez, L., Piattini, M. (2000d). Measuring the Quality of Entity Relationship Diagrams. Entity-Relationship 2000, Salt Lake City, November.

Gray, R., Carey, B., McGlynn, N. and Pengelly A. (1991). Design metrics for database systems. *BT Technology*, 9(4).

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

ISO/IEC 9126-1.2 (1999). Information technology- Software product quality – Part 1: Quality model.

Kesh, S. (1995). Evaluating the Quality of Entity Relationship Models. *Information and Software Technology*, 37 (12), 681-689.

Kitchenham, B., Pflegger, S. and Fenton, N. (1995). Towards a Framework for Software Measurement Validation. IEEE Transactions of Software Engineering, 21(12), 929-943.

Krogstie, J., Lindland, O.I. and Sindre, G. (1995). Towards a Deeper Understanding of Quality in Requirements Engineering, *Proceedings of the 7th International Conference on Advanced Information Systems Engineering (CAISE)*, Jyvaskyla, Finland, 82-95.

Li, H. and Cheng, W. (1987). An empirical study of software metrics. *IEEE Transactions on Software Engineering*, 13 (6), 679-708.

Lindland, O., Sindre, A. and Solvberg, A. (1994). Understanding Quality in Conceptual Modeling. *IEEE Software*, March, 11(2), 42-49.

McClure, C. (1992) *The Three R's of Software Automation: Re-engineering, Repository, Reusability*. Englewood Cliffs: Prentice-Hall.

Melton, A. (1996). *Software Measurement*, London, International Thomson Computer Press.

Moody, L. (1998). Metrics For Evaluating the Quality of Entity Relationship Models. *Proceedings of the Seventeenth International Conference on Conceptual Modelling (ER '98)*, Singapore, November 16-19.

Morasca, S. and Briand, L. (1997). Towards a Theoretical Framework for measuring software attributes. Proceeding of the Fourth International, Software Metrics Symposium, 119-126.

Muller, R. (1999). *Database Design For Smarties: Using UML for Data Modeling.* Morgan Kaufman.

Piattini, M., Caballero, I., Genero, M. and Calero, C. (2000). Data Quality and Database Desing. Circuits, Systems, Communications & Computers, July, Greece. del 2000. El

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

Schneidewind, N. (1992). Methodology For Validating Software Metrics. *IEEE Transactions of Software Engineering*, 18(5), 410-422.

Teorey, T. (1999). *Database modeling and design*. Third edition. San Francisco, Morgan Kaufmann.

Van Den Berg and Van Den Broek (1996). Axiomatic Validation in the Software Metric Development Process (Chapter 10). *Software Measurement.* Edited by Austin Melton. Thomson Computer Press.

Weyuker, E.J. (1988). Evaluating software complexity measures. *IEEE Transactions on Software Engineering.* 4(9), 1357-1365.

Whitmire, S.A. (1997), Object Oriented Design Measurement, Ed. Wiley.

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