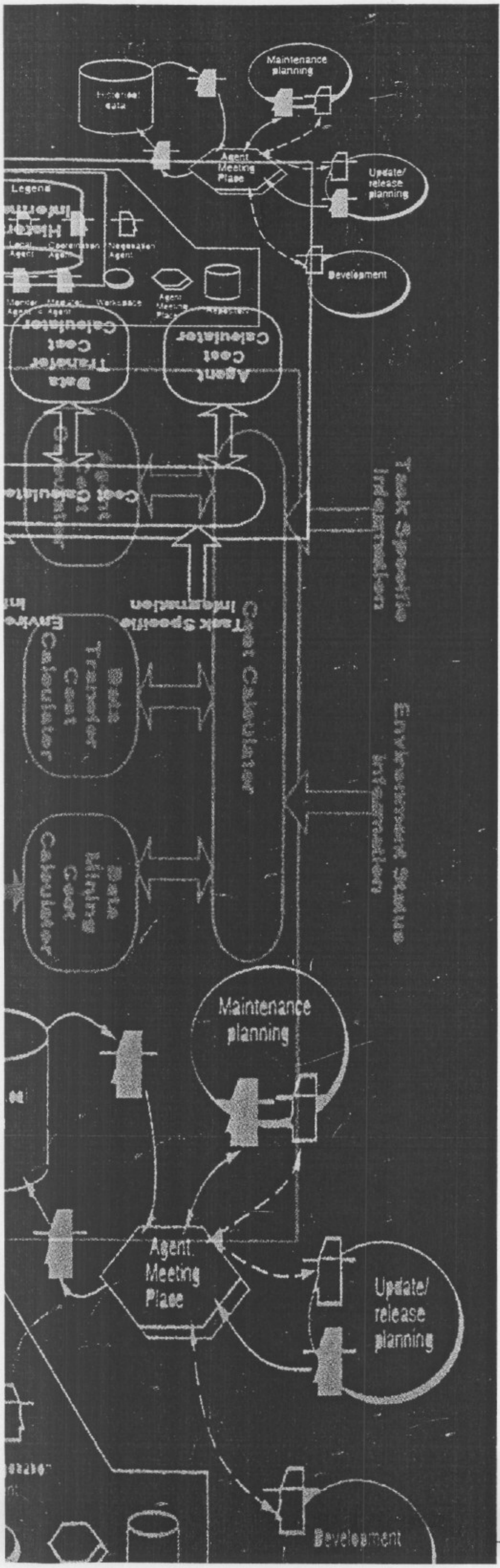


# SEKE

## proceedings

June 13-15, 2001

Buenos Aires, Argentina

the 13th international conference on software engineering & knowledge engineering

printed by
Knowledge Systems Institute

# PROCEEDINGS

# SEKE 2001

## Thirteenth International Conference on Software Engineering and Knowledge Engineering

Sponsored by

Knowledge Systems Institute, USA

Co-Sponsored by

\gencia Nacional de Promocisn Cientmfica y Tecnolsgica, Argentina
Branden University, USA
Universidad Nacional de Santiago del Estero, Argentina
Universidad Politicnica de Madrid, Spain

Technical Program
June 13-15, 2001
Buenos Aires, Argentina

Organized by

Knowledge Systems Institute

ISBN 1-891706-06-3 (paper)

Additional Copies can be ordered from:

Knowledge Systems Institute
3420 Main Street
Skokie, IL 60076, USA
Tel: +1-847-679-3135
Fax: +1-847-679-3166
Email: office@ksi.edu
http://www.ksi.edu

Printed in the United States of America

# SEKE 2001

## Thirteenth International Conference on Software Engineering and Knowledge Engineering

### Conference Co-Chairs

Natalia Juristo, Universidad Politicnica de Madrid, Spain
Steven Mellor, Project Technology, Inc., USA

### Program Co-Chairs

Silvia T. Acuqa, Universidad Nacional de Santiago del Estero, Argentina
Marcelo Estayno, Universidad Nacional de La Matanza, Argentina

### Steering Committee

Vic Basili, University of Maryland, USA
Bruce Buchanan, University of Pittsburgh, USA
Shi-Kuo Chang, University of Pittsburgh and Knowledge Systems Institute, USA
C. V. Ramamoorthy, University of California, USA

### Program Committee

Juan C. Augusto, Argentina
Paul Bailes, Australia
Mario R. Barbacci, USA
Richard Benjamins, Spain
Antonia Bertolino, Italy
Leopoldo Bertossi, Chile
Kai H. Chang, USA
William Cheng-Chung Chu, Taiwan
Reidar Conradi, Norway
Yi Deng, USA
Andreas Dengel, Germany
Khaled El Emam, Canada
Norberto Ezquerra, USA
Miguel Felder, Argentina
Dieter Fensel, The Netherlands
Alfonso Fuggetta, Italy
Kokichi Futatsugi, Japan
David Garlan, USA
Mari Georges, France
Robert L. Glass, USA
Claude, France
Christiane Gresse von Wangenheim, Brazil
Frank Houdek, Germany
Christian S. Jensen, Denmark
Marta Lspez, Spain
Chung-Horng Lung, Canada
Josi Carlos Maldonado, Brazil
Roland Mittermeir, Austria
Sandro Morasca, Italy
Ana M. Moreno, Spain
John Mylopoulos, Canada
Markku Oivo, Finland
Amy T. Parra, USA
Derek Partridge, UK
Shari L. Pfleeger, USA
Angel Puerta, USA
Ahmed A. Rafea, Egypt
David S. Robertson, Scotland
Guenther Ruhe, Germany
Carolyn B. Seaman, USA
Marcello A. Visconti, Chile
Jeffrey M. Voas, USA
Lawrence Votta, USA
Claes Wohlin, Sweden
Vilas Wuwongse, Thailand

### Local Arrangements Committee

Diana Burlando, Argentina
Oscar Dieste, Spain
Jorge Eterovic, Argentina
Ricardo Imbert, Spain
Cecilia Lasserre, Argentina
Ramsn Ledesma, Argentina
Alfredo M. Olivero, Argentina
Hugo Rene Padovani, Argentina
Mabel Sosa, Argentina
Sira Vegas, Spain

### Finance Chair

Judy Pan, USA

# PROGRAM CO-CHAIRS' MESSAGE

Welcome to the Thirteenth International Conference on Software Engineering and Knowledge Engineering! On behalf of the Conference Committees and Chairs, we would like to express our pleasure that you are here to share this event with us. 17 countries from all five continents are represented in our Program. By the end of the Conference, you should have a good understanding of the status of research and applications worldwide in the field of interactions between Software Engineering and Knowledge Engineering.

The worldwide response to the Call for Papers was more than expected (taking into account how far away Argentina is from almost anywhere). We received more than 100 contributions. This meant that we could select high-quality submissions. After a rigorous review process, around 48 full papers and 8 short papers were selected for presentation at the conference.

SEKE'01 follows on the earlier success of twelve efforts to bring together these two disciplines. This year the conference is organized in 16 sessions in a double track format, including a special session for short papers. Papers cover different topics, including Software Quality, Reuse, Requirements Engineering, Databases, Object Orientation, Project Management, Software Agents, Learning Organizations, Software Design and Maintenance, Software Architectures and Data Mining. In these areas, there are papers focusing on the intersection of SE and KE, as well as on issues addressed from a SE&KE perspective and from more traditional perspectives.

We are fortunate to have five prominent keynote speakers: Steve Mellor, Vic Basili, Shari Pfleeger, Al Davis and Shi-Kuo Chang (in order of appearance).

We are very grateful for having had the opportunity to serve as program Co-chairs for this conference. The organization of a major conference is a collective effort involving many people. We would like to thank the many individuals and organizations who have contributed to the success of this conference. Natalia Juristo, who trusted us regarding the feasibility of holding SEKE'01 in Argentina and was responsible for proposing Buenos Aires to the SEKE Steering Committee, deserves a special mention. Natalia and the members of the Software Engineering Group at the Universidad Politécnica de Madrid in Spain have given us support, advice and have worked extremely hard to make SEKE'01 a success. The commitment of Sira Vegas, who has not stinted any effort with regard to conference organization, is especially noteworthy.

Silvia T. Acuña and Marcelo Estayno
SEKE'01 Program Co-Chairs

# Table of Contents

## TRACK A. Object Orientation



## TRACK B. Project Management



## TRACK A. Software Agents

**TRACK B. Short Papers**



**TRACK A. Learning Organizations**

# Knowledge Discovery For Predicting Entity Relationship Diagram Maintainability

Marcela Genero, José A. Olivas, Mario Piattini, Francisco Romero
Department of Computer Science
University of Castilla-La Mancha
Ronda de Calatrava, 5
13071, Ciudad Real (Spain)
E-mail: {mgenero, jaolivas, mpiattin, fpromero}@inf-cr.uclm.es

## Abstract

It is generally accepted that the quality of the information system (IS) is highly dependent on decisions made early in the IS development. Conceptual data model is a key artifact built at the early phases of the IS life cycle, therefore its quality influence on the quality of IS which is finally implemented. We focus this paper on the maintainability of conceptual data models, because it is one of the most crucial quality characteristic. As maintainability is an external quality attribute that can only be measured once an IS is finished or nearly finished, our idea is to present a set of measures for measuring the structural complexity of Entity Relationships diagram (ERD), whose values can be obtained at the early phases of the IS life cycle, and based on these metrics values we will be able to predict ERD maintainability. For building the prediction model, we have used an extension of the original Knowledge Discovery in Databases (KDD): the Fuzzy Prototypical Knowledge Discovery (FPKD) that consists of the search for fuzzy prototypes that characterise the maintainability of an ERD. These prototypes lay the foundation of the prediction model that will lead us to predict ERD maintainability.

**Keywords:** entity relationship diagram maintainability, entity relationship metrics, information system quality, maintainability prediction, knowledge discovery, fuzzy deformable prototypes

## 1. Introduction

The demand for increased IS quality has made quality more of a discriminating factor between software products than ever before. It is generally accepted that the quality of the IS is highly dependent on decisions made early in the development. The construction of conceptual data models is often an important task of this early development. The conceptual data model lays the foundation of all later design work, so its quality has a significant impact on the quality of the IS which is ultimately implemented. Therefore, improving the quality of conceptual data models will be a major step towards the quality improvement of the IS development. We will focus on ERD because in today's database design world it is still the dominant method of conceptual modelling [1].

The definition of the different characteristics that compose the concept of "quality" of an ERD is not enough on its own in order to ensure quality in practice, as people will generally make different interpretations of the same concept. Software measurement plays an important role in this sense because metrics provide a valuable and objective insight into specific ways of enhancing each of the software quality characteristics.

Quality is a multidimensional concept, composed of different characteristics such as functionality, reliability, usability, efficiency, maintainability and portability [2]. We focus our paper on maintainability, because maintainability is one of the most crucial quality characteristic in software development organisations. However maintainability is an external quality attribute that can only be measured once an IS is finished or nearly finished, so it is highly important to avail of metrics for measuring an internal quality attribute, such as complexity, early in the IS life cycle, and after predict maintainability based on those metrics.

The availability of significant measures in the early phases of the IS life cycle allows for better management of the later phases, and more effective quality assessment when quality can be more easily affected by corrective actions. If these measures, applied in early phases of the development process, are found to be useful, they can be used to detect problems in the early design before implementation starts, thus potentially saving time and effort for rework on the design.

Some of the metric proposals that have appeared for measuring conceptual data model quality are the following: Gray [3], Eick [4], Kesh [5], Moody [6] and Genero et al. [7]. We based this work on the Genero et al. 's proposal because it provides objective metrics which were theoretically validated [7] and also partially empirically validated [8]. Genero et al. [7] have presented the theoretical validation of these metrics following Zuse's framework, demostrating that they they are characterised by the ratio scale type.

Genero et al. [8] have presented a case study, concluding that a great correlation exists between the proposed metrics for measuring ERD structural complexity and the time spent in the maintenance of the application programs that manage the data graphically represented by the ERD. As empirical validation is critical to the success of software measurement [9-12], we are aware that it is neccesary to do further validation. Therefore we have carried out a controlled experiment, with the objective of predicting each of the sub-characteristics of ERD maintainability (understandability, simplicity, analysability, modifiability, stability, and testability ) at the initial phases of the IS life cycle. For analysing data collected in the experiment, we have used an extension of the original KDD [13]: the FPKD [14, 15] that consists of the search for fuzzy prototypes [16] that characterise the maintainability of an ERD. These prototypes lay the foundation of the prediction model that will lead us to predict ERD maintainability.

This paper is organised in the following way: In section 2 we present a set of metrics for measuring ERD structural complexity [7]. In section 3 we describe a controlled experiment carried out in order to use the metrics presented in section 2 for predicting ERD maintainability. Also in section 3 we present an extension of the original KDD process, the FPKD used to build the prediction model. In section 4 we show how to predict ERD maintainability. Lastly, in section 5 we will summarise the paper, draw our conclusions, and present the future trends in the field of data models metrics for assuring quality in conceptual modelling.

## 2. Metrics for ERD structural complexity

In this section we present a set of comprehensive and objective metrics for measuring the structural complexity of ERD. These metrics have been proposed in [7].

- NE METRIC. Is the total number of entities within an ERD.
- NA METRIC. Is the total number of attributes that exist within the ERD. In this metric, all types of attributes, such as simple attributes, composite attributes and also multivalued attributes, take the value 1.
- DA METRIC. Is the total number of derived attributes existing in an ERD.
- CA METRIC. Is the total number of composite attributes within an ERD.
- MVA METRIC. Is the total number of multivalued attributes within an ERD.
- NR METRIC. Is the total number of relationships within the ERD. In this metric we consider all types of relationships, such as, common relationships, generalisations/specialisations and aggregations. In the case of generalisations/specialisation we consider one relationship for each pair child-parent. In the case of aggregation relationships we consider one relationship for each pair whole-part.
- 1:NR METRIC. Is the total number of 1:N (including 1:1) relationships within an ERD.
- M:NR METRIC. Is the total number of M:N relationships within an ERD.
- N-ARYR METRIC. Is the total number of N-ary relationships (not binary) within an ERD.
- BINARYR METRIC. Is the total number of binary relationships within an ERD.
- NIS_AR METRIC. Is the total number of IS_A relationships (generalisation/specialisation) that exist within an ERD. In this case, we consider one relationship for each pair child-parent within the IS_A relationship.
- NAggR. Is the total number of aggregations in an ERD. In this case, we consider one relationship for each pair whole-part within the aggregation relationship.
- RR METRIC. Is the total number of relationships that are redundant in an ERD. It is calculated with the help of expert database designers, they have to decide which relationships are redundant.
- SD METRIC. We define SD metric thus:

$$SD = \sum_{i=1}^{N^e} a_i^2$$

$a_i$ is the number of entities which can be directly reached from the entity "i" through relationships.
$N^e$ is the number of entities in aERD.

For calculating this metric we only consider common relationships (neither generalisation nor aggregation). In the case of reflexive relationships, between an entity and itself, we consider the value 1.

These metrics are open-ended metrics [17], i.e. they are not bounded in an interval. Also could be useful close-ended metrics (percentage metrics), such as the following: RR/NR; DA/NA; M.NR/NR; 1:NR/NR, BinaryR/NR, N-AryR/NR, etc...

We have theoretically validated these metrics [7] following the formal measurement framework proposed by Zuse [18], with the objective of ascertaining the scale type of each metric, concluding that all of these open-ended metrics are in ratio scale and the close-ended metrics are in absolute scale.

## 3. A comprehensive controlled experiment for predicting ERD maintainability

Taking into account some suggestions provided in [19, 20] about how to do empirical studies in Software Engineering, we carried out a controlled experiment with the aim of predicting ERD maintainability based on metric values (see section 2) obtained at the early phases of IS life cycle.

### 3.1 Subjects

The experimental subjects used in this study were: 9 professors and 7 students enrolled in the final-year of Computer Science in the Department of Computer Science at the University of Castilla-La Mancha in Spain. All of the professors belong to the Software Engineering area and they have on average five years of experience in the design of ERD. By the time the experiment was done all of the students had had two courses on Software Engineering and one course on Databases, in which they learnt in depth how to build ERD. Moreover, subjects were given an intensive training session before the experiment took place.

### 3.2 Experimental materials and tasks

The subjects were given twenty seven ERD [21, 22] of different universes of discourse, but general enough to be easily understood by each of the subjects. Each diagram has a test enclosed which includes the description of maintainability sub-characteristics, such as: understandability, simplicity, analysability, modifiability, stability and testability. Each subject has to rate each sub-characteristic using a scale consisting of seven linguistic labels. For example for understandability we proposed the following linguistic labels:

| | | | | | | |
|---|---|---|---|---|---|---|
| Extremely difficult to understand | Very difficult to understand | A bit difficult to understand | Neither difficult nor easy to understand | Quite easy to understand | Very easy to understand | Extremely easy to understand |

We allowed one week to do the experiment, i.e., each subject had carry out the test alone, and could use unlimited time to solve it. After completion of the tasks subjects were asked to complete a debriefing questionnaire. This questionnaire included (i) personal details and experience, (ii) opinions on the influence of different components of ERD, such as: entities, relationships, attributes, etc... on their maintainability.

### 3.3 Experimental design and data collection

The INDEPENDENT VARIABLES are those metrics proposed in section 2.

The DEPENDENT VARIABLES are those of the maintainability sub-characteristics: understandability, simplicity, analysability, modifiability, stability and testability, measured according to the subject's rating.

We decided to give our subjects as much time as they needed to finish the test they had to carry out. All tests were considered valid because all of the subjects have at least medium experience in building ERDs (this fact was corroborated analysing the responses of the debriefing questionnaire).

### 3.4 Construction of fuzzy deformable prototypes to characterise ERD maintainability

We have used an extension of the original KDD [13]: the FPKD [14, 15] that consists of the search for fuzzy prototypes [16] that characterise the maintainability of an ERD. In the rest of this section we will explain each of the steps we have followed in the FPKD process.

#### *3.4.1 Selection of the target data*

We have taken as a start set a relational database that contains 432 records (with 15 fields, 9 represent metrics values, 6 represent maintainability sub-characteristics) obtained from the calculation of the metric values (for each ERD) and the responses of the experiment given by the subjects.

### *3.4.2 Preprocessing*

The Data-Cleaning was not necessary because we didn't find any errors.

### *3.4.3 Transformation*

This step was performed doing different tasks:

- SUMMARISING SUBJECT RESPONSES. We built a single table with 27 records (one record for each model) each with 17 fields (1 of ERD number, 1 of prototype type, 9 of metrics and 6 of maintainability sub-characteristics). This table is shown in appendix A. The metric values were calculated measuring each diagram, and the values for each maintainability sub-characteristics were obtained aggregating subjects´ ratings using the mean of them. We only consider 9 of the 14 proposed in section 2, because the examples we considered all of the other metrics take zero value.
- CLUSTERING BY REPERTORY GRIDS. In order to detect the relationships between the ERD, for obtaining those which are easy, medium or difficult to maintain (based on subjects' ratings of each maintainability sub-characteristics), we have carried out a hierarchical clustering process by Repertory Grids. The set of elements is constituted by the 27 ERD, the constructions are the intervals of values of the subjects´rating. To accomplish an analysis of clusters on elements, we have built a proximity matrix that represents the different similarities of the elements, a matrix of 27 x 27 elements (the diagrams) that above the diagonal represents the distances between the different diagrams. Converting these values to percentages, a new table is created and the application of Repertory Grids Analysis Algorithm returns a graphic as a final result (see figure 1).

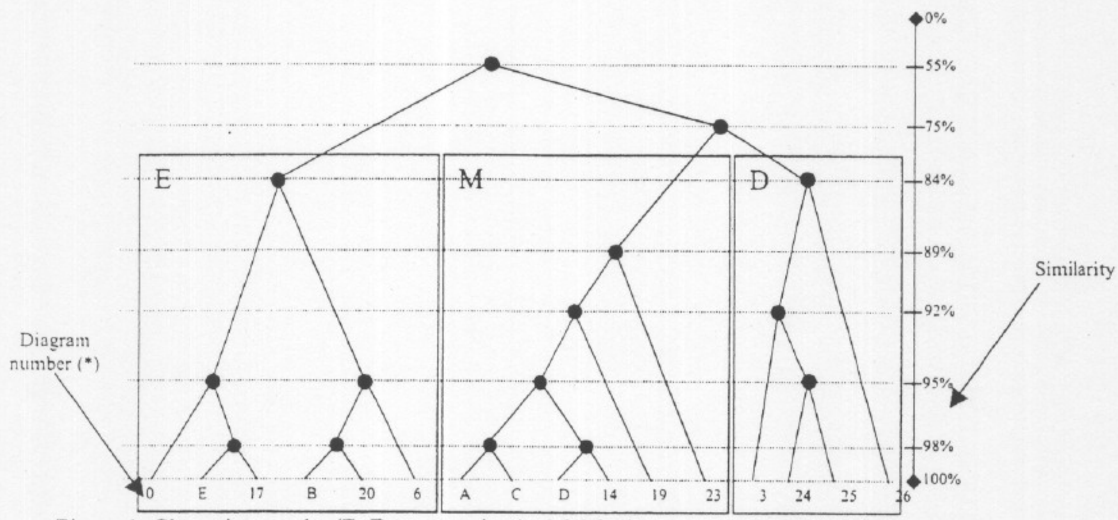


Figure 1. Clustering results (E: Easy to maintain, M: Medium to maintain, D: Difficult to maintain)
(*) We have grouped some ERDs assigning them one letter because they have 100% of similarity (see appendix A)

- DATA MINING. The selected algorithms for data mining process were summarise functions. Table 1 shows the parametric definition of the prototypes. These parameters will be modified taking into account the degree of affinity of a new ERD with the prototypes. With the new modified prototype we will be able to predict the maintainability of a new ERD.

| **Difficult** | Understandability | Simplicity | Analisability | Modificability | Stability | Testability |
|---|---|---|---|---|---|---|
| Average | 6 | 6 | 6 | 5 | 6 | 6 |
| Max. | 6 | 6 | 6 | 5 | 6 | 7 |
| Min. | 5 | 5 | 5 | 5 | 5 | 5 |
| **Medium** | | | | | | |
| Average | 3 | 4 | 4 | 4 | 4 | 4 |
| Max. | 4 | 4 | 4 | 4 | 5 | 4 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Min. | 3 | 4 | 3 | 3 | 3 | 4 |
| **Easy** | | | | | | |
| Average | 3 | 3 | 3 | 3 | 3 | 3 |
| Max. | 3 | 3 | 3 | 3 | 4 | 3 |
| Min. | 2 | 2 | 2 | 2 | 2 | 2 |

Table 1. Prototypes "Easy, Medium and Difficult to maintain"

- FORMAL REPRESENTATION OF CONCEPTUAL PROTOTYPES. The prototypes have been represented as fuzzy numbers, which are going to allow us to obtain a degree of membership in the concept. For the sake of simplicity in the model, they have been represented by triangular fuzzy numbers. Therefore, in order to construct the prototypes (triangular fuzzy numbers) we only need to know their centrepoints ("centre of the prototype"), which are obtained by normalising and aggregating the metric values corresponding to the ERDs of each of the prototypes (see figure 2).

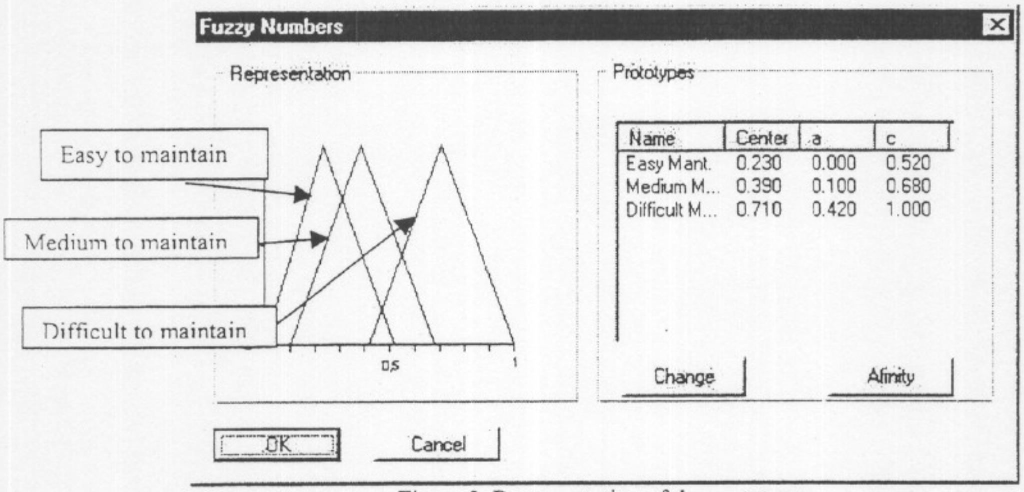


Figure 2. Representation of the prototypes

### 3.5 Threats to Validity

We will discuss the empirical study's various threats to validity and the way we attempted to alleviate them:

- THREATS TO CONSTRUCT VALIDITY. We propose subjective metrics for measuring each of the dependent variables (maintainability sub-characteristics) based on the judgement of the subjects (see section 3.2). As the subjects involved in this experiment have medium experience in ERD design we think their ratings could be considered significant. The independent variables (each of the metrics proposed in section 2) that measure the structural complexity of an ERD can also be considered constructively valid, because from a system theory point of view, a system is called complex if it is composed of many (different types of elements), with many (different types of) (dynamically changing) relationships between them [23].
- THREATS TO INTERNAL VALIDITY. Seeing the results of the experiment we can conclude that empirical evidence of the existing relationship between the independent and the dependent variables exists. We have tackled different aspects that could threaten the internal validity of the study, such as: differences among subjects, knowledge of the universe of discourse among ERD diagrams, accuracy of subject responses, learning effects, fatigue effects, persistence effects and subject motivation.
- THREATS TO EXTERNAL VALIDITY. Two threats to validity have been identified which limit the ability to apply any such generalisation, and we

tried to alleviate them: materials and tasks and subject selection.

In general in order to extract a final conclusion we need to replicate this experiment with a greater number of subjects, including practitioners. After doing replication we will have a cumulative body of knowledge; which will lead us to confirm if the presented metrics could really be used as early quality indicators, and could be used to predict ERDs maintainability.

## 4. Prediction of ERD maintainability

Using Fuzzy Deformable Prototypes [14, 15], we can deform the most similar prototype to a new ERD, and define the factors for a new situation, using a linear combination with the degrees of membership as coefficients. We will show an example of how to deform the fuzzy prototypes found in section 3.4.

Given the metric values corresponding to a new ERD (see table 2) and their normalised values (see table 3), the final average is 0.69, and the affinity with prototypes is shown in figure 3.

| NE | NA | NR | M:NR | 1:NR | N_ARY | BINARY_R | NIS_AR | SD |
|---|---|---|---|---|---|---|---|---|
| 11 | 30 | 9 | 4 | 5 | 2 | 7 | 7 | 65 |

Table 2. Metric values for a new ERD

| NE | NA | NR | MN_R | 1:NR | N-ARY | BINARY_R | NIS_AR | SD |
|---|---|---|---|---|---|---|---|---|
| 0.8 | 0.7 | 0.7 | 0.5 | 0.5 | 1.0 | 0.6 | 0.8 | 0.9 |

Table 3. Normalised values of metrics shown in table 2

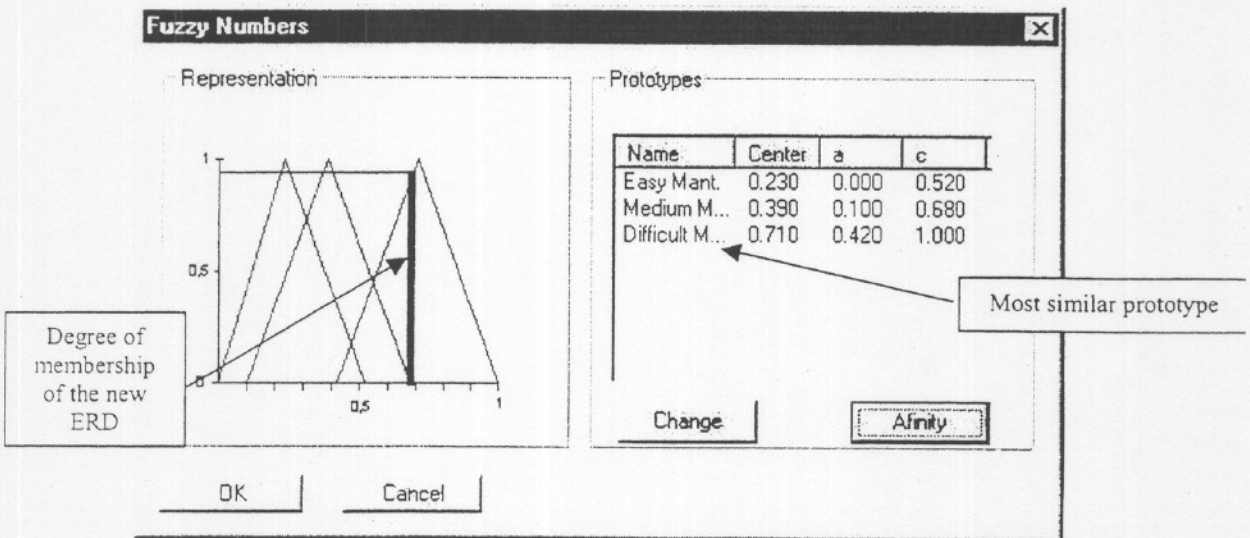


Figure 3. Affinity of the real case with the prototypes

The most similar prototype for this new ERD is "Difficult to maintain", with a degree of membership of 0.94. Then, the prediction is shown in table 4.

| | Understandability | Simplicity | Analisability | Modifiability | Stability | Testability |
|---|---|---|---|---|---|---|
| Average | 6 | 6 | 6 | 5 | 6 | 6 |
| Maximum | 6 | 6 | 6 | 5 | 6 | 7 |
| Minimum | 5 | 5 | 5 | 5 | 5 | 5 |

Table 4. Predicted maintainability sub-characteristics values for a new ERD

We want to highlight that this a first approach to predict ERD maintainability, we need "real data" about ERD maintainability efforts, like time spent in maintenance tasks in order to predict data that can be highly useful to software designers and developers.

## 5. Conclusions and future work

Due to the growing complexity of IS and the relevant role that data itself have in an IS, continuous attention to and assessment of conceptual models is necessary to produce quality IS.

As the aim of this paper is to predict ERD maintainability early in the IS life cycle, we have presented a set of metrics for assessing the structural complexity of ERD, proposed by Genero et al. [7]. In order to empirically validate each of the proposed metrics as maintainability indicators We have also carried out a controlled experiment, and from the empirical data obtainded we have built prediction model for maintainability sub-characteristics (understandability, simplicity, analysability, modifiability, stability, and testability ). The prediction model was built following an extension of the traditional KDD called FPKD. This model has also been used for different kinds of real problems, such as forest fire prediction, financial analysis or medical diagnosis, with very good results [14].

Nevertheless, despite the encouraging results obtained we are aware that we need to do more metric validation, both empirical and theoretical in order to assess if the presented metrics could really be used as early quality indicators.

We cannot disregard the increasing diffusion of the object-oriented paradigm in conceptual modelling. We think that object oriented models are more appropriate than ERD to describe the kind of IS built nowadays. We have also been working on metrics for measuring OMT [24] class diagrams [25] and also UML [26] class diagrams [27].

Furthermore, we will not only address the maintainability sub-characteristics, we also have to focus our research on measuring other quality factors as proposed in the ISO 9126 [2].

## Acknowledgements

This research is part of the MANTICA project, partially supported by CICYT and the European Union (1FD1997-0168TIC), the DOLMEN project supported by CICYT (TIC 2000-1673-C06-06) and the CIPRESES project supported by CICYT (TIC 2000-1362-C02-02).

## References


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

[2] ISO/IEC 9126-1 (1999). Information Technology-Software product quality – Part 1: Quality Model.

[3] Gray, R., Carey, B., McGlynn, N. and Pengelly A. (1991). Design metrics for database systems. *BT Technology*, 9(4), 69-79.

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

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

[6] Moody, L. (1998). Metrics for Evaluating the Quality of Entity Relationship Models. *Proceedings of the Seventeenth International Conference on Conceptual Modelling (E/R '98)*, Singapore, November 16-19, 213-225.

[7] Genero, M., Piattini, M. and Calero, C. (2000). Measures to get better quality databases. *2nd Itenational Conference on Enterpriese Information Systems - ICEIS 2000*, Stafford, 49-55.

[8] Genero, M., Piattini, M. and Calero, C. (2000). An Approach To Evaluate The Complexity Of Conceptual Database Models. *3rd European Software Measurement Conference - FESMA 2000*, Madrid.

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

[10] Fenton, N. and Pfleeger, S. (1997). *Software Metrics: A Rigorous Approach*. 2nd. edition. London, Chapman & Hall.

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

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

[13] Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996). The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Communications of the ACM*, 39(11), 27 – 34.

[14] Olivas, J. A. and Romero, F. P. FPKD. (2000). Fuzzy Prototypical Knowledge Discovery. Application to Forest Fire Prediction. *Proceedings of the SEKE'2000*, Knowledge Systems Institute, Chicago, Ill. USA, 47 – 54.

[15] Olivas, J. A. (2000). Contribution to the Experimental Study of the Prediction based on Fuzzy Deformable Categories, PhD Thesis, University of Castilla-La Mancha, Spain.

[16] Zadeh, L. (1982). A note on prototype set theory and fuzzy sets. *Cognition* 12, 291 – 297.

[17] Lethbridge, T. (1998). Metrics For Concept-Oriented Knowledge bases. *International Journal of Software Engineering and Knowledge Engineering*, 8(2), 161-188.

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

[19] Briand, L., Arisholm, S., Counsell, F., Houdek, F., and Thévenod-Fosse, P. (1999). Empirical Studies of Object-Oriented Artifacts, Methods, and Processes: State of the Art and Future Directions. *Technical Report IESE 037.99/E, Fraunhofer Institute for Experimental Software Engineering*, Kaiserslautern, Germany.

[20] Briand, L., Bunse, C. and Daly, J. (1999). A Controlled Experiment for evaluating Quality Guidelines on the Maintainability of Object-Oriented Designs. *Technical Report IESE 002.99/E, Fraunhofer Institute for Experimental Software Engineering*, Kaiserslautern, Germany.

[21] Ruiz, I. and Gómez-Nieto, M. (1997). *Diseño y uso de Bases de Datos Relacionales*. Ra-Ma. (In Spanish).

[22] De Miguel, A. and Piattini, M. (1999). *Fundamentos y Modelos de Bases de Datos*. 2nd ed. Ra-Ma. (In Spanish).

[23] Poels, G. and Dedene, G. Measures for Assessing Dynamic Complexity Aspects ofObject-Oriented Conceptual Schemes. (2000). *In: Proceedings of the 19th International Conference on Conceptual Modeling (ER 2000)*, Salt Lake City, USA, 499-512.

[24] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1991). *Object- Oriented Modeling and Design*. Prentice Hall. USA.

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

[26] OMG. Object Management Group. *UML Revision Task Force. (1999). OMG Unified Modeling Language Specification, v. 1.3. document ad/99-06-08.*

[27] Genero, M., Piattini, M. and Calero, C. (2000) Early Measures For UML class diagrams. *L'Objet*. 6(4), Hermes Science Publications, 489-515.

## Appendix A

The following table summarises the data collected in the experiment described in section 3. The first row identifies each ERD number. Attached to some diagrams appear one letter. The diagrams which have the same letter mean that they have 100% of similarity. The second column presents the type of the prototype to which each ERD diagram belongs (E=Easy to maintain; M=medium to mantain; D=Difficult to maintain), the next nine columns show the metric values, and the last columns show each maintainability sub-characteristic rate (S1=Understandability; S2=Simplicity; S3=Analisability; S4=Modifiability; S5= Stability; S6=Testability).

| | Prototype | NE | NA | NR | M:NR | 1:NR | N-AryR | Binary_R | NIS_AR | SD | S1 | S2 | S3 | S4 | S5 | S6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ERD 0 | E | 5 | 19 | 5 | 2 | 3 | 0 | 5 | 0 | 19 | 2 | 2 | 2 | 3 | 3 | 3 |
| ERD 1 (A) | M | 12 | 44 | 11 | 3 | 8 | 2 | 9 | 4 | 42 | 4 | 4 | 4 | 4 | 4 | 4 |
| ERD 2 (A) | M | 6 | 23 | 7 | 0 | 7 | 0 | 7 | 2 | 21 | 4 | 4 | 4 | 4 | 4 | 4 |
| ERD 3 | D | 9 | 33 | 11 | 5 | 6 | 0 | 11 | 2 | 60 | 5 | 5 | 5 | 5 | 5 | 5 |
| ERD 4 (B) | E | 13 | 15 | 11 | 8 | 3 | 0 | 11 | 5 | 32 | 3 | 3 | 3 | 3 | 3 | 3 |
| ERD 5 (C) | M | 6 | 16 | 4 | 4 | 0 | 1 | 3 | 3 | 44 | 3 | 4 | 4 | 4 | 4 | 4 |
| ERD 6 | E | 6 | 16 | 40 | 4 | 0 | 1 | 3 | 3 | 36 | 2 | 3 | 3 | 3 | 4 | 3 |
| ERD 7 (A) | M | 11 | 33 | 8 | 1 | 7 | 0 | 8 | 4 | 27 | 4 | 4 | 4 | 4 | 4 | 4 |
| ERD 8 (D) | M | 6 | 45 | 7 | 0 | 7 | 0 | 7 | 0 | 30 | 3 | 4 | 3 | 4 | 4 | 4 |
| ERD 9 (E) | E | 4 | 2 | 3 | 0 | 3 | 0 | 3 | 0 | 10 | 2 | 2 | 2 | 2 | 3 | 2 |
| ERD 10 (D) | M | 9 | 31 | 6 | 1 | 5 | 0 | 6 | 2 | 20 | 3 | 4 | 3 | 4 | 4 | 4 |
| ERD 11 (B) | E | 6 | 17 | 3 | 0 | 3 | 0 | 3 | 2 | 8 | 3 | 3 | 3 | 3 | 3 | 3 |
| ERD 12 (C) | M | 12 | 11 | 5 | 4 | 1 | 0 | 5 | 9 | 16 | 3 | 4 | 4 | 4 | 4 | 4 |
| ERD 13 (D) | M | 10 | 35 | 11 | 4 | 11 | 1 | 10 | 0 | 75 | 3 | 4 | 3 | 4 | 4 | 4 |
| ERD 14 | M | 8 | 29 | 6 | 0 | 6 | 0 | 6 | 2 | 19 | 3 | 4 | 3 | 4 | 3 | 4 |
| ERD 15 (B) | E | 6 | 29 | 4 | 4 | 0 | 2 | 2 | 0 | 46 | 3 | 3 | 3 | 3 | 3 | 3 |
| ERD 16 (C) | M | 6 | 34 | 5 | 2 | 3 | 0 | 5 | 0 | 22 | 3 | 4 | 4 | 4 | 4 | 4 |
| ERD 17 | E | 6 | 15 | 5 | 0 | 5 | 0 | 5 | 0 | 25 | 2 | 2 | 2 | 2 | 2 | 2 |
| ERD 18 (E) | E | 6 | 16 | 4 | 2 | 2 | 0 | 4 | 0 | 20 | 2 | 2 | 2 | 2 | 3 | 2 |
| ERD 19 | M | 8 | 19 | 5 | 5 | 0 | 0 | 5 | 3 | 18 | 3 | 4 | 3 | 3 | 4 | 4 |
| ERD 20 | E | 5 | 10 | 3 | 2 | 1 | 0 | 3 | 1 | 12 | 3 | 3 | 3 | 3 | 4 | 3 |
| ERD 21 (B) | E | 9 | 26 | 5 | 2 | 3 | 0 | 5 | 4 | 13 | 3 | 3 | 3 | 3 | 3 | 3 |
| ERD 22 (C) | M | 8 | 12 | 5 | 3 | 2 | 0 | 5 | 4 | 25 | 3 | 4 | 4 | 4 | 4 | 4 |
| ERD 23 | M | 11 | 32 | 7 | 1 | 6 | 0 | 7 | 4 | 35 | 4 | 4 | 4 | 4 | 5 | 4 |
| ERD 24 | D | 11 | 36 | 10 | 6 | 4 | 1 | 9 | 4 | 37 | 6 | 5 | 6 | 5 | 6 | 5 |
| ERD 25 | D | 11 | 41 | 12 | 6 | 6 | 1 | 11 | 8 | 40 | 5 | 6 | 6 | 5 | 6 | 5 |
| ERD 26 | D | 12 | 45 | 12 | 5 | 7 | 2 | 10 | 8 | 67 | 6 | 6 | 6 | 5 | 6 | 7 |