APAQS 2001

Proceedings of the Second Asia-Pacific Conference on Quality Software

Hong Kong, December 10-11, 2001

Edited by Y.T. Yu and T.Y. Chen

ORGANIZED BY:

- DEPARTMENT OF COMPUTER SCIENCE, CITY UNIVERSITY OF HONG KONG, HONG KONG
- CENTRE FOR SOFTWARE ENGINEERING, SCHOOL OF INFORMATION TECHNOLOGY, SWINBURNE UNIVERSITY OF TECHNOLOGY, AUSTRALIA

CO-ORGANIZED BY:

- INFORMATION TECHNOLOGY SERVICES DEPARTMENT, THE GOVERNMENT OF THE HONG KONG SAR

SPONSORED BY:

- THE BRITISH COMPUTER SOCIETY HONG KONG SECTION
- HONG KONG COMPUTER SOCIETY
- THE HONG KONG INSTITUTION OF ENGINEERS INFORMATION TECHNOLOGY DIVISION
- IEEE HONG KONG SECTION COMPUTER SOCIETY CHAPTER

Table of Contents

Second Asia-Pacific Conference on Quality Software – APAQS 2001

Keynote Address I

Professor C.V. Ramamoorthy, University of California, Berkeley, USA

Session 1A: Program Analysis I

Session 1B: Software Reliability

Session 3A: Formal Methods I

Session 3B: Testing I

Session 3C: Software Development II

Keynote Address II

Session 4A: Formal Methods II

Session 4B: Testing II

Session 4C: Process Models and Tools

Session 5A: Requirements Engineering

Session 5B: Usability

Session 5C: Process Improvement and Assessment

Session 6A: Formal Verification

Session 6B: E-commerce

Session 6C: Configuration Management

Please Note: Paper titles are preceded by the symbols [F], [S], and [E], respectively, to denote the categories of the papers: Full paper, Short paper, and Experience report.

Assurance of Conceptual Data Model Quality Based on Early Measures

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

Abstract

The increasing demand for quality information systems (IS), has become quality the most pressing challenge facing IS development organisations. In the IS development field it is generally accepted that the quality of an IS is highly dependent on decisions made early in its development. Given the relevant role that data itself plays in an IS, conceptual data models are a key artifact of the IS design. Therefore, in order to build "better quality" IS it is necessary to assess and to improve the quality of conceptual data models based on quantitative criteria. It is in this context where software measurement can help IS designers to make better decision during design activities. We focus this work on the empirical validation of the metrics proposed by Genero et al. for measuring the structural complexity of entity relationship diagrams (ERDs). Through a controlled experiment we will demonstrate that these metrics seem to be heavily correlated with three of the sub-factors that characterise the maintainability of an ERD, such as understandability, analysability and modifiability.

Keywords: information system quality, conceptual data model, entity relationship diagram, structural complexity, metrics, empirical validation

1. Introduction

The increasing demand for quality information systems (IS), has become quality the most pressing challenge facing any IS development organisation. The quality of an IS is highly dependent on decisions made early in its development. The IS design activities are generally divided into two groups: those related to database issues and those related to operational issues. Therefore, the databases constitute only one of the components of an IS, which also includes application programs, user interfaces and service programs. However the central role that the data itself plays in an IS more than justifies an independent study of database conceptual design, and its contribution to overall IS quality. Given that conceptual data modelling is a first step in database conceptual design, the quality of conceptual data models will greatly influence the database quality and also the quality of the IS which is finally delivered.

Bear in mind the points previously mentioned, in order to build "better quality" IS it is necessary to improve the quality of conceptual data models. However, before improving their quality it is necessary to assess it in an objective way. It is in this context where software measurement can help IS designers to make better decision during design activities.

The early availability of metrics allows IS designers:

1. a quantitative comparison of design alternatives, and therefore and objective selection among several ERD alternatives with equivalent semantic content.
2. a prediction of external quality characteristics, like maintainability in the initial phases of the IS life cycle and a better resource allocation based on these predictions.

Even though, several quality frameworks for conceptual data models have been proposed [1, 2, 3, 4], most of them lack valid quantitative measures to evaluate the quality of conceptual data models in an objective way. Papers referring to metrics for conceptual data models are scarce. Kesh [5] has proposed a set of metrics for ERD, but they are theoretically based, and of limited applicability in practice. Moody [6] has proposed a set of 25 metrics to measure different characteristics of conceptual data models. Some of them are objectively calculated while others are subjective, based only on experts´rating, and moreover their utility in practice has not been demonstrated. Genero et al. [7] have proposed a set of metrics for measuring the ERD structural complexity, arguing that ERD structural complexity could heavily influence ERD maintainability and as a result

0-7695-1287-9/01 $17.00 © 2001 IEEE

could affect the overall IS quality. Their theoretical validation was presented in [7] following Zuse´s framework [8], demonstrating that most of these metrics are in the ratio scale. Also they were partially empirically validated in [9], demonstrating by means of a case study that they seem to be heavily correlated with the time spent on the different phases of the development of the application programs that manage the data represented in the ERD.

In any case given that empirical validation is critical to the success of software measurement [10, 11, 12, 13] and following their authors suggestion they must be put through further empirical validation. For this reason we decided to carry out this study. Therefore, the main objective of this work is to assess through experimentation, if some of the available metrics for measuring ERD structural complexity are valid metrics that could be used as early quality indicators. In this way the IS designers will have a valid measurement support that will guide their designs.

This work is organised in the following way: the ERD structural complexity metrics proposed by Genero et al. [7] are presented in section 2. In section 3 we present a controlled experiment carried out for ascertaining if any relationship exists between the metrics and three of the maintainability sub-characteristics [14]: understandability, analysability and modifiability. Finally in section 4, we present some concluding remarks and future trends in metrics for conceptual modeling.

2. Measures for ERD structural complexity

In this section we present the definition of Genero et al.´s metrics [7] for measuring ERD structural complexity. As the structural complexity of an ERD is determined by the different elements that compose it, such as entities, attributes, relationships, generalisations, etc., these metrics are classified into the following categories: entity metrics, attribute metrics and relationship metrics.

2.1 Entity metrics

– NE METRIC. The Number of Entities metric is defined as the total the number of entities within an ERD.

2.2 Attribute metrics

– NDA METRIC. The Number of Derived Attributes metric is defined as the total number of derived attributes within an ERD.

– NCA METRIC. The Number of Composite Attributes metric is defined as the total number of composite attributes within an ERD.

– NMVA METRIC. The Number of Multivalued Attributes metric is defined as the total number of multivalued attributes within an ERD.

– NA METRIC. The Number of Attributes metric is defined as the total number of attributes that exist within an ERD, taking into account both entity and relationship attributes. In this number we include simple attributes, derived attributes, composite attributes and also multivalued attributes, each of which take the value 1 when we calculate this metric.

2.3 Relationship metrics

– NR METRIC. The Number of relationships metric is defined as the total number of relationships within an ERD, taking into account only common relationships.

– NM:NR METRIC. The Number of M:N Relationships metric is defined as the total number of M:N relationships within an ERD.

– N1:NR METRIC. The Number of 1:N relationships metric is defined as the total number of 1:N relationships (including also 1:1 relationships) within an ERD.

– NN-ARYR METRIC. The Number of N-ary Relationships metric is defined as the total number of N-ary relationships (not binary) within an ERD.

– NBINARYR METRIC. The Number of Binary relationships metric is defined as the total number of binary relationships within an ERD.

– NIS_AR METRIC. The Number of IS_A Relationship metric is defined as the total number of IS_A relationships (generalisation/specialisation) within an ERD. In this case, we consider one relationship for each child-parent pair within the IS_A relationship.

– NREFR METRIC. The Number of Reflexive Relationships metric is defined as the total number of reflexive relationships within an ERD.

– NRR METRIC. The Number of Redundant Relationships metric is defined as the total number of redundant relationships within an ERD.

– SD METRIC. The Schema Cohesion metric is defined as:

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

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 within an ERD.

These metrics are open-ended metrics [15], i.e. they are not bounded in an interval. Close-ended metrics (percentage metrics) could also be useful, such as the following: NRR/NR; NDA/NA; NM.NR/NR; N1:NR/NR, NBinaryR/NR, NN-AryR/NR, etc.

3. A comprehensive controlled experiment

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

Taking into account some suggestions provided about how to perform successful empirical studies in software engineering [16, 17, 18, 19], we carried out a controlled experiment with the aim testing the following hypotheses:

Hypothesis 1:

- Null hypothesis, H_0: There is no significant correlation between ERD structural complexity measures and the subjects´rating of understandability, analysability and modifiability.
- Alternative hypothesis, H_1: There is a significant correlation between ERD structural complexity measures and the subjects´rating of understandability, analysability and modifiability.

Hypothesis 2:

- Null hypothesis, H_0: There is no significant correlation between ERD structural complexity measures and the understandability time.
- Alternative hypothesis, H_1: There is a significant correlation between ERD structural complexity measures and the understandability time.

Considering that maintainability is one of the factors which influences quality [12], if we finally show that there is some relation, we will be also showing the usefulness of these metrics in assessing and controlling ERD quality.

3.1 Subjects

The experimental subjects used in this study were: 36 students enrolled in the third year of Computer Science at the Department of Computer Science at the University of Castilla-La Mancha in Spain. By the time the experiment was done all of them had one course 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 nine ERDs related to the same universe of discourse (Stock Exchange). The structural complexity of each diagram is different, because the values of the measures are different for each diagram (see table 2).

Each diagram had a test enclosed which includes two parts:

- Part 1: A questionnaire in order to evaluate if the subjects really understand the content of the ERDs. Each questionnaire contained exactly the same number of questions (five) and the questions were conceptually similar and in identical order. Each subject had to write down the time spent answering the questionnaire, by recording the initial time and final time. The difference between the two is what we call the understandability time (expressed in minutes).
- Part 2: consists of a definition of each of the three maintainability sub-characteristics, such as: understandability, analysability, modifiability. Each subject has to rate each sub-characteristic using a scale consisting of seven linguistic labels. For example for understandability we proposed the linguistic labels shown in table 1.

Table 1. Linguistic labels for understandability

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 associate a number between 1 and 7 to each linguistic label: The worst case takes the value seven (extremely difficult to understand), and the best case takes the value one (extremely easy to understand).

The subjects were given all the materials described above. We explained to them how to carry out the tests. We allowed one hour to do all the tests. Each subject had to work alone. In case of doubt, they could only consult the supervisor who organised the experiment.

3.4 Experimental design and data collection

The INDEPENDENT VARIABLE is the structural complexity of ERDs, measured by means of the proposed metrics (see section 2).

The DEPENDENT VARIABLES are three of the maintainability sub-characteristics: understandability, analysability, modifiability, measured according to the subject's rating. The understandability is also measured in an objective way, by means of the time the subjects spent answering the questionnaire, called the understandability time.

We selected a within-subject design experiment, i.e. all the tests (i.e. experimental tasks) had to be solved by each of the subjects. The subjects were given the tests in different order.

We collected all the data, including the initial and final time of the first part of the test, and the subjects´ratings of the second part. The measure values for each ERD were automatically calculated by means of the MANTICA Tool [20].

We collected all the tests controlling if they were complete and the responses were correct. We discarded the tests of 9 subjects, because they included an incorrect answer. Therefore, we take into account the responses of 27 subjects.

3.5 Experiments Results

We summarise subject responses in a single table (see table 1) with 9 rows (one row for each ERDs) and 14 columns (the first column represent the identification of each ERD, the next 9 columns represent the metric values and the last four columns represent the maintainability sub-characteristics: F1 = understandability, F2 = analysability and F3 = modifiability, and the understandability time). The values for each maintainability sub-characteristics and the understanbadibility time, that appear in table 2 were obtained aggregating the data collected using their mean.

Table 2. Summary of data collected in the experiment

ERDs	METRIC VALUES									Average of subjects´s rating			Understan-dability time (minutes)
	NE	NA	NR	N1:NR	NM:NR	NBinary_R	NN-AryR	NRefR	SD	F1	F2	F3	
ERD 1	2	2	6	2	0	2	0	0	2	2.59	3.04	2.89	3.20
ERD 2	5	15	5	5	0	5	0	0	16	3.19	3.26	3.37	3.40
ERD 3	8	27	9	9	0	9	0	0	32	3.89	4.04	4.11	4.18
ERD 4	11	45	15	12	3	13	2	3	119	4.04	4.73	4.77	4.25
ERD 5	12	38	7	5	2	5	2	0	63	3.26	3.74	3.96	3.00
ERD 6	13	54	17	14	3	15	2	3	130	4.65	5.08	5.08	4.60
ERD 7	7	30	5	5	0	4	1	0	28	3.00	3.35	3.65	3.67
ERD 8	13	55	17	14	3	15	2	3	137	4.63	4.78	5.22	4.80
ERD 9	15	41	9	6	3	7	2	0	88	4.00	4.52	4.74	4.26

We used the data shown in table 2 to test the hypotheses formulated at the begging of section 3.

First, we applied the Kolmogrov-Simonorov test to ascertain if the distribution of the data collected was normal or not. As the data was nonormal we decided to use a nonparametric test like Spearman´s correlation coefficient, with a level of significance $\alpha = 0.05$, which means the level of confidence is 95% (i.e. the probability that we reject H_0 when H_0 is false is at least 95%, which is statistically acceptable).

Using Spearman´s correlation coefficient, each of the metrics was correlated separately to subject´s ratings about understandability, analysability and modifiability, and also to understandability time (see table 3).

Table 3. Spearman´s correlation between the proposed metrics and understandability, analysability, modifiability and the understandability time

Metrics	Understandability	Analysability	Modifiability	Understandability Time
NE	0.812	0.828	0.899	0.678
NA	0.900	0.933	0.957	0.767
NR	0.937	0.937	0.884	0.810
N1:NR	0.962	0.962	0.885	0.894
NM:NR	0.967	0.867	0.879	0.730
NBinary_R	0.975	0.950	0.906	0.874
NN-AryR	0.727	0.764	0.798	0.484
NRefR	0.822	0.822	0.766	0.730
SD	0.950	0.967	0.983	0.800

Analysing the Spearman´s correlation coefficients shown in table 2, we can conclude that there is a high correlation (rejecting hypothesis H_0) between the understandability time and the modifiability time and the metrics NE, NA, NR, N1:NR, NM:NR, NbinaryR, NN-aryR, NRefR and SD. There also exists a high correlation between these metrics and the understandability time. We can deduce this because the correlation coefficient is greater than 0.6, which is a common threshold to evaluate correlation values..

Even though the sample size (nine diagrams) 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 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 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.

As the diffusion of the experimental data is important to external replication [21] of the experiments we have put all of the material of this experiment in the web http:\\alarcos.inf-cr.uclm.es.

Moreover it is necessary to perform more experimentation using data directly obtained from "real cases", as for example ERD maintainability time, in order to draw final conclusions.

3.5 Threats to Validity

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

3.5.1. Threats to Construct Validity

The construct validity is the degree to which the independent and the dependent variables accurately measure the concepts they purport to measure. The dependent variables we used are maintainability sub-characteristics: understandability, analysability and modifiability. We propose subjective metrics for them (using linguistic variables), based on the judgement of the subjects (see section 3.2). As the subjects involved in this experiment have medium experience in ERD design, and taking into account that the tasks they must to perform were not difficult, we think their ratings could be considered significant. The time spent by the subjects answering the questionnaire is also a significant indicator of the time they need to really understand each diagram.

For construct validity of the independent variables, we have to address the question to which degree the metrics used in this study measure the concept they purport to measure. Our idea is to use metrics presented in section 2 to measure the structural complexity of an ERD. 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 [22]. According to this, we think that the construct validity of our independent variables can thus be considered satisfactory. In spite of this, we consider that more experiments must be done, in order to draw a final conclusion to assure construct validity.

3.5.2 Threats to Internal Validity

The internal validity is the degree to which conclusions can be drawn about the causal effect of independent variables on the dependent variables. The following issues have been dealt with:

- DIFFERENCES AMONG SUBJECTS. Using a within-subjects design, error variance due to differences among subjects is reduced. As Briand et al. [18] remarks in software engineering experiments when dealing with small samples, variations in participant skills are a major concern that is difficult to fully address by randomisation or blocking. In this experiment, all of the students had the same degree of experience in modelling with ERD.

- KNOWLEDGE OF THE UNIVERSE OF DISCOURSE. All of the ERDs are related to the same universe of discourse (Stock Exchange), and it is general enough to be easily understood by each of the subjects. Therefore, the knowledge of the domain does not threaten to the internal validity.
- PRECISION IN THE TIME VALUES. The subjects were responsible for recording the start and finish times of each test. We think this method is more effective than having a supervisor who records the time of each subject. However, we are aware that the subjects could introduce some imprecision.
- LEARNING EFFECTS. The subjects were given the tests in different order, to cancel out learning effects. Subjects were required and controlled to answer in the order in which the tests appeared.
- FATIGUE EFFECTS. On average the experiment lasted for less than one hour, so fatigue was not very relevant. Also, the different order in the tests helped to cancel out these effects.
- PERSISTENCE EFFECTS. In order to avoid persistence effects, the experiment was run with subjects who had never done a similar experiment.
- SUBJECT MOTIVATION. We motivated students to participate in the experiment, explaining to them that similar tasks to the experimental ones could be done in exams or practice by students, so they wanted to take the most of the experiment.
- OTHER FACTORS. Plagiarism and influence between students were strictly controlled. Students were told that talking to each other was forbidden and the professor who carried out the experiment controlled them.

Seeing the results of the experiment we can conclude that it seems that there is evidence of relationship between the independent and the dependent variables. However, only by replicating controlled experiments, where the measures would be varied in a controlled manner and all factors would be kept constant, really be demonstrated causality.

3.5.3. Threats to External Validity

The external validity is the degree to which the results of the research can be generalised to the population under study and other research setting. The greater the external validity, the more the results of an empirical study can be generalised to actual software engineering practice. Two threat of validity have been identified which limit the ability to apply any such generalisation:

- MATERIALS AND TASKS USED. In the experiment we tried to use ERDs and tasks which can be representative of real cases, but more empirical studies taking "real cases" from software companies must be done.
- SUBJECTS. We are aware that more experiments with practitioners and professionals must be carried out in order to be able to generalise these results. However, in this case, the tasks to be performed do not require high levels of industrial experience, so, experiments with students could be appropriate [13].

4. Conclusions

We have presented a set of objectives and automatically well-defined metrics which were proposed in [9] with the objective of measuring ERD structural complexity. As early available metrics, they will mainly be useful in two senses:

- To help IS designers choose design alternatives
- To predict ERD maintainability from the early stages of IS design

We have shown how we carried out an empirical study through a controlled experiment, corroborating that most of the proposed metrics (NE, NA, NR, NM:NR, N1:NR, NBinaryR, NN-AryR, NRefR, SD) have a high correlation with the understandability, analysability and modifiability of and ERD, and also with the understandability time.

Nevertheless, despite the encouraging results obtained we are aware that we need to do more metric validation in order to assess if the proposed metrics could be really help IS designers take better decisions in their design tasks, which is the most important goal of any measurement proposal that aims to be useful [23].

Our final goal is to predict ERD external quality attributes, such as maintainability. Based on the metrics proposed for ERD structural complexity we are building a prediction model [24] for ERD maintainability. We have to do further work in order to evaluate the prediction accuracy of our model.

We cannot disregard the increasing diffusion of the object-oriented paradigm in conceptual modelling. Modern approaches towards OO system development, like Catalysis [25] and Rational Unified Process [26] consider conceptual modelling as a key step in the development life-cycle. Object-oriented (OO) models are really more suitable than ERD describing the kind of IS built nowadays. In relation to OO models, we have also been working on metrics for measuring UML [27] class diagrams [28]. All of these studies are related to of OO measures for conceptual models focused on static diagrams, but as was remarked in several works [22,29],

[22] there is a need for measuring dynamic diagrams, like state diagrams, activity diagrams, etc.

Furthermore, we will not only address the maintainability sub-characteristics, but we also have to focus our research measuring other quality factors as proposed in the ISO 9126[14].

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).

References

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

[2] L. Moody, and Shanks G., "What Makes A Good Data Model? Evaluating The Quality of Entity Relationships Models", *Proceedings of the 13th International Conference on Conceptual Modelling (ER '94)*, Manchester, England, 1994, pp. 94-111.

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

[4] R. Schuette, and T. Rotthowe, "The Guidelines of Modeling – An Approach to Enhance the Quality in Information Models", *Proceedings of the Seventeenth International Conference on Conceptual Modelling (ER '98)*, Singapore, 1998, pp. 240-254.

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

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

[7] M. Genero, M. Piattini, and C. Calero, "Measures to get better quality databases", *2nd Iternational Conference on Enterprise Information Systems - ICEIS 2000*, Stafford, 2000, pp. 49-55.

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

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

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

[11] N. Fenton, and S. Pfleeger, *Software Metrics: A Rigorous Approach*, 2nd. Edition, Chapman & Hall, London, 1997.

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

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

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

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

[16] D. Perry, A. Porter, and L. Votta, "Empirical Studies of Software Engineering: A Roadmap", *Future of Software Engineering*. Ed:Anthony Finkelstein, ACM, 2000, pp. 345-355.

[17] L. Briand, S. Arisholm, F. Counsell, F. Houdek, and P. Thévenod-Fosse, "Empirical Studies of Object-Oriented Artifacts, Methods, and Processes: State of the Art and Future Directions", *Empirical Software Engineeering*, 2000.

[18] L. Briand, C. Bunse, and J. Daly, "A Controlled Experiment for evaluating Quality Guidelines on the Maintainability of Object-Oriented Designs", *IEEE Transactions on Software Engineering*, 27(6), 2001, pp. 489-515.

[19] L. Briand, J. Wüst, J. Daly, and D. Porter, "Exploring the relationships between design measures and software quality in object-oriented systems", *The Journal of Systems and Software 51*, 2000, pp.245-273.

[20] M. Genero, M. Piattini, J. Rincón-Cinca, and M. Serrano, "MANTICA: Una Herramienta de Métricas para Modelos de Datos", *CACIC 2000*, Usuahia, Argentina, 2000. (in spanish).

[21] A. Brooks, J. Daly, J. Miller, M. Roper, and M. Wood, "Replication of experimental results in software engineering", *Technical report ISERN-96-10*, International Software Engineering Research Network, 1996.

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

[23] N. Fenton, and M. Neil, "Software Metrics: a Roadmap". *Future of Software Engineering*, Ed:Anthony Finkelstein, ACM, 2000, pp. 359-370.

[24] M. Genero, J. Olivas, M. Piattini, and F. Romero. *"Knowledge Discovery For Predicting Entity Relationship Diagram Maintainability"*, *SEKE 2001*, Argentina, Knowledge Systems Institute, 2001, pp. 203-211.

[25] D. D'Souza, and A. Wills, *Objects, Components and Frameworks with UML: the Catalysis Approach*, Addison-Wesley, 1999.

[26] Rational Software. Object Oriented Analysis and Design, Student Manual. http://www.rational.com/, 1998.

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

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

[29] F. Brito e Abreu, H. Zuse, H. Sahraoui, and W. Melo, "Quantitative Approaches in Object-Oriented Software Engineering", *Object-Oriented technology: ECOOP'99 Workshop Reader*, Lecture Notes in Computer Science 1743, Springer-Verlag, 1999, pp. 326-337.