QSSE 2004

The Proceedings of the Fourth ASERC Workshop on Quantitative and Soft Computing based Software Engineering

February 16-17, 2004

Banff, Alberta, Canada

QSSE 2004

The Proceedings of the Fourth ASERC Workshop on Quantitative and Soft Computing based Software Engineering

February 16-17, 2004

Banff, Alberta, Canada

Organized by the Alberta Software Engineering Research Consortium (ASERC)
and the Department of Electrical and Computer Engineering, University of Alberta

Organization Committee

General Chair:

Petr Musílek, University of Alberta

Program Chair:

Witold Pedrycz, University of Alberta

Finance Chair:

Marek Reformat, University of Alberta

Publication Chair:

Scott Dick, University of Alberta

Keynotes Chair:

Lukasz Kurgan, University of Alberta

Organization Chair:

Yifan Li, University of Alberta

Focus of the workshop:

The focus of the workshop and relevant papers is on the development and deployment of software systems. In addition, papers describing the introduction and improvement of metric programs, sharing of project data, and working towards improvement of software processes with the goal of leveraging quality and effectiveness of software production between academic departments and industry were reviewed by the committee, including work in-progress and position papers.

Table of Contents

[redacted]

Preface

Welcome to QSSE 2004, the 4th ASERC Workshop on Quantitative and Soft Computing Based Software Engineering.

The phenomenal growth of complex software systems in both number and size has caused Software Engineering to face an increasingly difficult task. It is no longer feasible to expect engineers to have a complete mental model of the systems they are attempting to build in software. The quantitative approach to Software Engineering has been steadily gaining more attention as the complexity of software products and processes become omnipresent in current practice. The traditional tools for handling quantitative data, such as statistics, are often not adequate when the data are incomplete, sparse or, on the other hand, too extensive and highly dimensional. This is where methods of Soft Computing can be used to get the most of the data and interpret the results.

The main intent of the workshop is to bring researchers in these fields together in a joint effort of pursuing innovation in Software Engineering. The initiative of the Alberta Software Engineering Research Consortium (ASERC), under whose auspices this workshop is held, is extremely helpful in this regard.

The keynote presentations delivered by experts in the area focus on research milestones of this enormously rich landscape of Software Engineering. The papers collected in the proceedings serve as a testimony to the rapid growth of the quantitative and soft facets in the field.

We hope that this workshop will stimulate research interaction within the vibrant research community in the province of Alberta and around the world. We are also confident that this workshop will foster stronger links among various academic groups – an essential component of successful Software Engineering research. Finally, we believe that the unique surroundings of the Banff National will stimulate fruitful discussions and help initiate interesting pursuits.

Welcome …

Petr Musílek
QSSE 2004 General Chair

Empirical Validation of Metrics for Datawarehouses

Manuel Serrano
Alarcos Research Group
University of Castilla – La Mancha
Paseo de la Universidad, 4
13071 Ciudad Real
(+34) 926 29 53 00
Manuel.Serrano@uclm.es

Coral Calero
Alarcos Research Group
University of Castilla – La Mancha
Paseo de la Universidad, 4
13071 Ciudad Real
(+34) 926 29 53 00
Coral.Calero@uclm.es

Mario Piattini
Alarcos Research Group
University of Castilla – La Mancha
Paseo de la Universidad, 4
13071 Ciudad Real
(+34) 926 29 53 00
Mario.Piattini@uclm.es

Juan Trujillo
Dept. de Lenguajes y Sistemas Informáticos
University of Alicante
P.O. Box 99. E-03080
jtrujillo@dlsi.ua.es

Sergio Luján
Dept. de Lenguajes y Sistemas Informáticos
University of Alicante
P.O. Box 99. E-03080
slujan@dlsi.ua.es

Abstract

Datawarehouses (DW), based on the multidimensional modeling, provide companies with huge historical information for the decision making process. As these DW's are crucial for companies in making decisions, their quality is absolutely critical. One of the main issues that influences their quality lays on the models (conceptual, logical and physical) we use to design them. In the last years, there have been several approaches to design DW's from the conceptual, logical and physical perspectives. However, from our point of view, there is a lack of more objective indicators (metrics) to guide the designer in accomplishing an outstanding model that allows us to guarantee the quality of these DW's. In this paper, we present a set of metrics to measure the quality of conceptual models for DW's. We have validated them through an empirical experiment performed by last course students in computer science. Our experiment showed us that several of the proposed metrics seems to be practical indicators of the quality of conceptual models for DW's.

Keywords

Datawarehouse quality. datawarehouse modeling, quality metrics, empirical validation.

1. Introduction

Datawarehouses (DW), based on the multidimensional modeling, provide companies with huge historical information for the decision making process [8]. As these DW's are crucial for companies in making decisions, their quality is absolutely critical.

One of the main issues that influence their quality lays on the models (conceptual, logical and physical) we use to design them. In the last years, different authors have suggested interesting recommendations for achieving a "good" datawarehouse data model. However, quality criteria are not enough on their own to ensure quality in practice, because different people will generally have different interpretations of the same concept. According to the Total Quality Management (TQM) literature, measurable criteria for assessing quality are necessary to avoid "arguments of style". The objective should be to replace intuitive notions of design "quality" with formal, quantitative measures in order to reduce subjectivity and bias in the evaluation process. However, for data modeling to progress from a "craft" to an engineering discipline, the desirable qualities of data models need to be made explicit.

However, from our point of view, there is a lack of more objective indicators (metrics) to guide the designer in accomplishing an outstanding model that allows us to guarantee the quality of these DW's.

The final aim of our work is to define a set of metrics for assuring data warehouse quality by means of measuring the data model quality.

Metrics definition must be done in a methodological way and it is necessary to follow a number of steps to ensure the reliability of the proposed metrics [4]. Obtaining a set of valid metrics is not only defining them. The process involves also the validation of the proposed metrics, in order to see if they are useful and valid. This validation should be made theoretical and empirically.

The theoretical validation helps us to know when and how apply the metrics [15] [11] and the goal of the empirical validation is to prove the practical utility of the proposed metrics [5] [9]. Although there are various ways of performing this step, basically we can divide the empirical validation into experimentation, case studies and surveys.

In the last years, we have been working on the definition of metrics for measuring the quality of datawarehouse models [4] [12][13]. We have made the theoretical validation of the metrics we have proposed and we are working on the empirical validation of those metrics for assuring their practical utility. In this paper we will show an experiment we have developed with a set of last course students of computer science.

The remain of the paper is structured as follows: Section 2 summarizes the conceptual model for DW's, based on the UML, which we will use as the framework to define our metrics. Section 3 defines the metrics for datawarehouse conceptual models we will use in our study. Section 4 shows the theoretical validation and section 5 describes the empirical validation we have performed with the proposed metrics. Finally, Section 6 draws conclusions and introduces future investigation arising from this work.

2. Object – oriented conceptual modelling with UML for datawarehouses

In this section, we outline our approach to datawarehouse conceptual modelling, based on the UML (Figure 1 shows an example of a datawarehouse conceptual model, specified in UML).

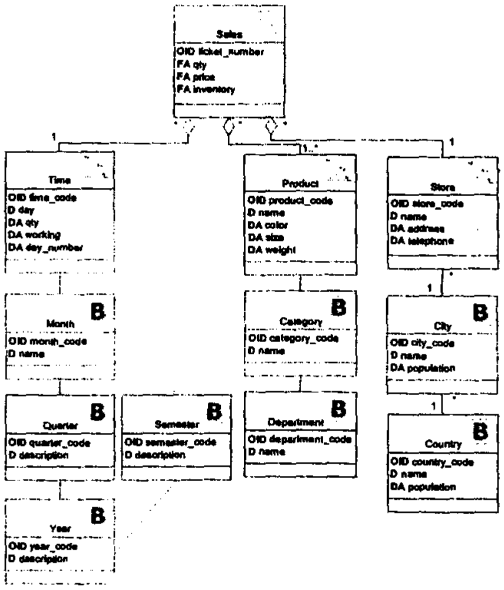

Figure 1. Example of an Object Oriented datawarehouse conceptual model using UML

This approach has been specified by means of a UML profile (A *profile* is a set of improvements that extend an existing UML type of diagram for a different use) that contains the necessary stereotypes in order to carry out conceptual modelling successfully [10]. The main features of multidimensional modelling considered are the relationships "many-to-many" between the facts and one specific dimension, degenerated dimensions, multiple classification and alternative path hierarchies, and the non strict and complete hierarchies. In this approach, the structural properties of multidimensional modelling are represented by means of a UML class diagram in which the information is clearly organized into facts and dimensions.

Facts and dimensions are represented by means of fact classes and dimension classes respectively. Fact classes are defined as composite classes in shared aggregation relationships of n dimension classes. A fact is composed of measures or fact attributes. With respect to dimensions, each level of a classification hierarchy is specified by a base class. An association of base classes specifies the relationship between two levels of a classification hierarchy.
We refer the reader to [14] [10] for a complete description of our approach.

3. Metrics for datawarehouse conceptual models

A metric definition should always be based on clear measurement goals. Metrics should be defined following organisation's needs that are related to external quality attributes. We must firstly specify the goals of the metrics we plan to create to follow our organization's needs, and then we state the derived hypotheses. In our particular context, the main goal is "Defining a set of metrics to assess and control the quality of conceptual datawarehouse schemas".

As Briand et al. [3] said, the structural properties (such as structural complexity) of a schema have an impact on its cognitive complexity (see figure 2). By cognitive complexity we mean the mental burden of the persons who have to deal with the artefact (e.g. developers, testers, maintainers and final users). High cognitive complexity leads an artefact to reduce their understandability and this conduce undesirable external quality attributes, such as decreased maintainability - a characteristic of quality; ISO 9126 [7].

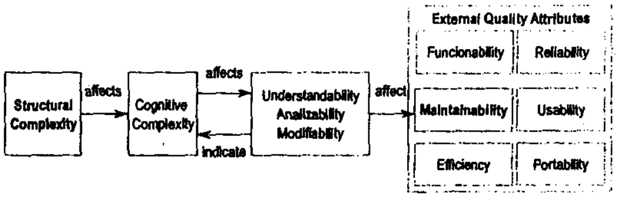

Figure 2. Relationship between structural properties, cognitive complexity, understandability and external quality attributes, based on [3]

Therefore, we can state our hypothesis as: "Our metrics (defined for capturing the structural complexity of a datawarehouse conceptual schema) can be used for controlling and assessing the quality of a datawarehouse (through its maintainability)".

Taking into account the metrics defined for datawarehouses at a logical level [12] and the metrics defined for UML class diagrams [6], we can propose an initial set of metrics for the model described in the previous section. When drawing up the proposal of metrics for datawarehouse models, we must take into account 3 different levels: class, star and diagram. In this papper we only show the metrics at diagram level (see table 1).

Table 1. Diagram scope metrics

Metric	Description
NFC	Number of Fact classes
NDC	Number of dimensional classes
NBC	Number of base classes
NC	Total number of classes NC = NFC + NDC + NBC
RBC	Ratio of base classes. Number of base classes per dimensional class
NSDC	Number of dimensional classes shared by more than one star
NAFC	Number of FA attributes of the fact classes
NADC	Number of D and DA attributes of the dimensional Tables.
NASDC	Number of D and DA attributes of the shared dimensional classes.
NA	Number of FA, D and DA attributes
NH	Number of hierarchies
DHP	Maximum depth of the hierarchical relationships
RDC	Ratio of dimensional classes. Number of dimensional classes per fact class.
RSA	Ratio of attributes. Number of FA attributes divided by the number of D and DA attributes.

Example

Table 2 summarizes the values for the defined metrics, regarding the example presented in the previous Section (figure 1).

Table 2. Star level metrics values

Metric	Value
NFC	1
NDC	3
NBC	8
NC	12
RBC	8/3
NSDC	0
NAFC	3
NADC	11
NABC	10
NASDC	0
NA	24
NH	3
DHP	3
RDC	3
RSA	3/21

4. Theoretical validation

Due to space constraints we cannot present the measure construction process for the proposed metrics for datawarehouse conceptual models using the [11]. In this process we have obtained that the metrics proposed are on a ratio scale. That means that they are formally valid software metrics because they are in the ordinal or in a superior scale, as remarked by Zuse [15], and are therefore perfectly usable.

5. Empirical validation

In this section, we will present our empirical validation for the metrics defined in the section 4. In doing this, we must firstly define the experimental settings (including the main goal of our experiment, the subjects that participated in the experiment, the main hypothesis under which we will run our experiment, the independent and dependent variables to be used in our model, the experimental design, the experiment running, material used and the subjects that performed the experiment). After that we discuss about the collected data validation. Finally, we analyse and interpret the results to find out if they follow the formulated hypothesis or not.

5.1. Experimental settings

Experiment goal definition

The goal definition of the experiment using the GQM approximation [1] can be summarized as:

To analyze the metrics for datawarehouse conceptual models
for the purpose of evaluating if they can be used as useful mechanisms
with respect of the datawarehouse maintainability
from the designer's *point of view*
in the context of last course students

Subjects

Twenty eight last course students participated in the experiment. The subjects were twenty three men and five, with an average age of 24.5 years. All the subjects have almost the same experience as they are all students.

Hypotheses formulation

The hypotheses of our experiment are:

Null hypothesis, H_0: There is no a statistically significant correlation between metrics and the maintainability of the schemas.
Alternative hypothesis, H_1: There is a statistically significant correlation between metrics and the maintainability of the schemas.

Alternative hypothesis H_1 is stated to determine if there is any kind of interaction between the metrics and the maintainability of a datawarehouse schema, based on the fact that the metrics are defined in an attempt to acquire all the characteristics of a conceptual datawarehouse model.

Variables in the study

Independent variables. The independent variables are the variables for which the effects should be evaluated. In our experiment these variables correspond to the metrics being researched. Table 3 presents the values for each metric in each schema.

Dependent variables. The maintainability of the tests was measured as the time each subject used to perform the tasks of each experimental test. The experimental tasks consisted in two different tasks, the former involves understanding the models by counting the number of classes that must be visited to access to a concrete information. The latter one involves the modification of the models to fit new design requirements.

Regarding time, it is necessary to point out that for each schema we separately record the understanding time (including understanding the model and the answering time to the first type of questions) and the modification time that includes the time spent in performing the second type of tasks.

Table 3. Values of the metrics for the schemas used in the experiment

	NDC	NBC	NC	RBC	NAFC	NADC	NABC	NA	NH	DHP	RSA
S01	6	16	23	2,67	1	7	9	17	6	4	0,06
S02	5	19	25	3,8	1	11	20	32	9	4	0,03
S03	2	5	8	2,5	4	4	6	14	3	2	0,4
S04	4	17	22	4,25	4	6	17	27	9	3	0,17
S05	3	21	25	7	4	8	24	36	7	4	0,13
S06	5	13	19	2,6	3	0	31	34	5	4	0,1
S07	3	6	10	2	3	7	2	12	5	2	0,33
S08	4	5	10	1,25	3	13	5	21	2	3	0,17
S09	3	5	9	1,67	2	12	5	19	2	3	0,12
S10	2	4	7	2	1	7	2	10	3	2	0,11

Material design and experiment running

Ten conceptual datawarehouse models were used for performing the experiment. Although the domain of the schemas was different, we tried to select representative examples of real world cases in such a way that the results obtained were due to the difficulty of the schema and not to the complexity of the domain problem. We tried to have schemas with different metrics values (see table 3).

We selected a within-subject design experiment (i.e. all the tests had to be solved by each of the subjects). The documentation, for each design, included a datawarehouse schema and a questions/answers form. The questions/answers form included the tasks that had to be performed and a space for the answers. For each design, the subjects had to analyse the schema, answer some questions about the design and perform some modifications on it.

Before starting the experiment, we explained to the subjects the kind of exercises that they had to perform, the material that they would be given, what kind of answers they had to provide and how they had to record the time spent solving the problems. We also explained to them that before studying each schema they had to annotate the start time (hour, minutes and seconds), then they could understand the design and answer the given question. Once the answer to the question was written, they had to annotate the final time (again in hour, minutes and seconds). Then they had to repeat the process with the modifications of the schema.

Tests were performed in distinct order by different subjects for avoiding learning and fatigue effects. The way we ordered the tests was using a randomisation function. To obtain the results of the experiment we used the number of seconds needed for each task on each schema by each subject.

5.2. Collected data validation

After marking the test, we obtained all the times for each schema and subject. We decided to study the outliers before working with the average data. In order to find the outliers we made a box plot (figures 3 and 4) with the collected data. Observing these box plots we can observe that there are several outliers. The outliers values were eliminated from the collected data. The descriptive statistics of the final set of data can be found in tables 4 and 5. Then, we perfomed the analysis with this data.

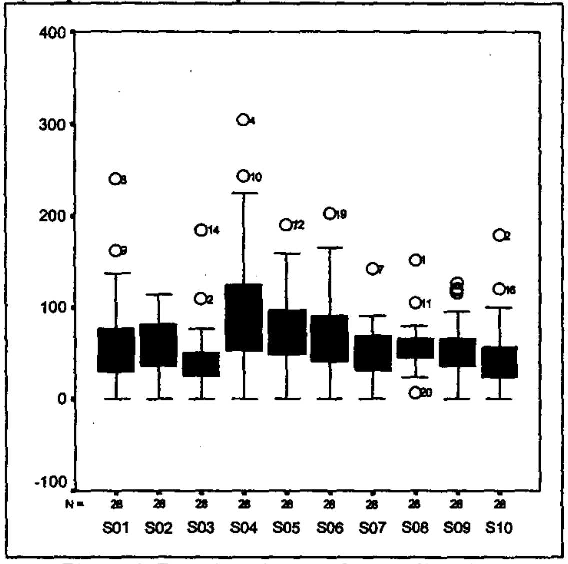

Figure 3. Box plot of the understanding time.

Table 4. Descriptive statistics of the understanding time

	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10
Avg	59,39	66,88	39,92	100,17	78,74	75,42	57,05	54,16	48,55	41,96
Min	25	21	19	47	31	26	24	24	15	13
Max	137	114	76	225	160	203	90	80	95	100
Dev	29,04	28,22	15,81	44,72	32,93	42,71	18,33	13,70	18,46	23,07

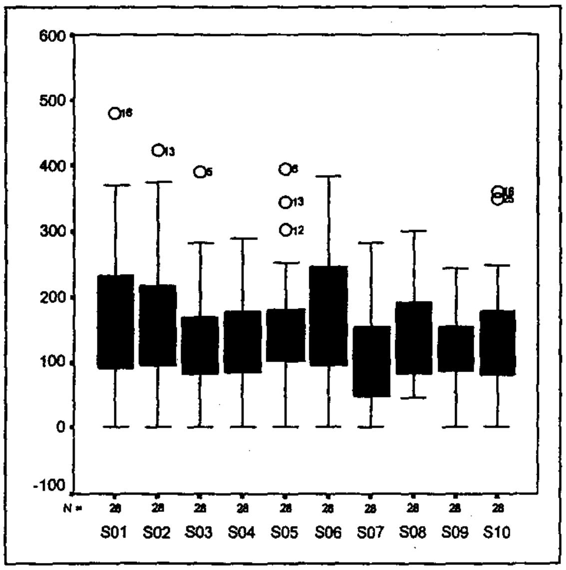

Fig. 1. Box plot of the modification time.

Table 5. Descriptive statistics of the modification time

	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10
Avg	172,71	166,46	129,83	143,76	138,95	177,60	119,17	140,75	121,00	126,44
Min	36	44	37	47	54	66	27	45	41	30
Max	371	374	282	289	253	383	283	300	244	247
Dev	87,71	78,33	57,21	65,00	45,74	88,90	67,94	65,87	54,79	55,83

Validity of results

As we know that different threats to the validity of the results of an experiment exist. In this section we will discuss threats to construct internal, external and conclusion validity.

Construct validity. The construct validity is the degree to which the independent and the dependent variables are accurately measured by the measurement instruments used in the study. The dependent variables we use, are understanding and modification times, i.e., the time each subject spent performing these tasks, so we consider these variables constructively valid. The construct validity of the measures used for the independent variables is guaranteed by the Distance framework [11] used for their theoretical validation.

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 should be considered:

- **Differences among subjects.** Within-subject experiments reduce variability among subjects.
- some way.
- **Precision in the time values.** The subjects were responsible believe this method is more effective than having a supervisor who records the time of each subject. However, we are aware that the subject could introduce some imprecision.
- **Learning effects.** Using a randomization function, tests were ordered and given in a distinct order for different subjects. So, each subject answered the tests in the given order. In doing this, we tried to minimize learning effects.
- **Fatigue effects.** The average time for completing the ... avoid these fatigue effects.
- **Persistence effects.** In our case, persistence effects are not present because the subjects had never participated in a similar experiment.
- **Subject motivation.** Subjects were volunteers and they were convinced that the exercises they were doing were useful. Therefore, we believe that subjects were motivated at doing the experiment.
- **Plagiarism and influence among subjects**. In order to avoid these effects a supervisor was present during the experiment. Subjects were informed they should not talk to each other or share answers with other subjects.

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

- **Materials and tasks used.** We tried to use schemas and operations representative of real world cases in the experiments, although more experiments with larger and more complex schemas are necessary.
- **Subjects.** Due to the difficulty of getting professionals to perform the experiment, the experiment was done by students. In general, more experiments with a larger number of subjects, students and professionals, and with a greater difference between the values of each metric are necessary to obtain more conclusive results.

Conclusion Validity. The conclusion validity defines the extent to which conclusions are statistically valid. The only issue that could affect the statistical validity of this study is the size of the sample data (17 values), which perhaps is not enough for both parametric and non-parametric statistic tests. We will try to obtain bigger sample data through more experimentation.

5.3. Analysis and interpretation

We used the data collected in order to test the hypotheses formulated previously. As we were not able to assure that the data we collected followed a common statistical distribution (mainly ... analysis, avoiding assumptions about the data normality. In this way, we made a correlation statistical analysis using the Spearman's Rho statistic and we used a level of significance $\alpha = 0.05$

Table 6 shows the results obtained for the correlation between each of the metrics and the time each subject used (on each schema) to perform the task of understanding. Table 7 shows the same data for the modification tasks.

Table 6. Results of the experiment (understanding time).

Metric	NDC	NBC	NC	RBC	NAFC	NADC	NABC	NA	NH	DHP	RSA

Table 7. Results of the experiment (modification time).

Metric	NDC	NBC	NC	RBC	NAFC	NADC	NABC	NA	NH	DHP	RSA
Corr.	0,84	0,52	0,62	0,50	-0,01	-0,26	0,77	0,60	0,49	0,78	-0,58
Sig.	0,00	0,12	0,06	0,14	0,97	0,46	0,01	0,07	0,15	0,01	0,08

Analysing both tables, we can conclude that there exists a high correlation between the understanding time used (understandability of the schemas) and the metrics NBC, NC, RBC, NABC, NA, NH and DHP (the value of significance is lower than α = 0.05). The other metrics do not seem to be correlated with the time. On the other hand, only the metrics NDC, NABC and DHP have correlation with modification time.

In considering these results, it seems that understandability is closer related to metrics that capture in some sense the "complexity" of the schemas. This complexity is captured by the number of classes of the schemas (size of the schema) and the number of hierarchy relationships in the stars. The modification time is not related to a great number of metrics, perhaps because the modification tasks could be solved focusing only on a small part of the schema.

Acknowledgements

This research is part of the CALIPO project, supported by Dirección General de Investigación of the Ministerio de Ciencia y Tecnologia (TIC2003-07804-C05-03). This research is also part of the MESSENGER project, supported by Consejeria de Ciencia y Tecnologia de la Junta de Comunidades de Castilla-La Mancha (PCC-03-003-1).

We would like to thank the students of fifth course of Computer Science from University of Castilla – La Mancha who kindly volunteered to take part in the experiment.

6. Conclusions

Data warehouses play a key role in the decision making process of companies, and therefore, assuring their quality is absolutely critical for this process. One way to achieve this quality objective is to assure the quality of the models (conceptual, logical and physical) used in designing them.

In this paper we have focused on validating empirically the proposed metrics for conceptual datawarehouse models, and we have presented an experiment we have accomplished. This experiment, which is the first approach to a complete empirical validation of the metrics, has showed us that it seems that there exist correlation between several of the metrics and the understandability of the conceptual datawarehouse models.

We are currently validating empirically all the proposed metrics, which will enable us to discard or refine these metrics. It would also be advisable to study the influence of the different analysis dimensions on the cognitive complexity of an object-oriented model; as well as the repercussion of using packages in the conceptual modelling of complex and extensive datawarehouses in order to simplify their design.

7. References

[1] Basili V. and Weiss D. A Methodology for Collecting Valid Software Engineering Data, *IEEE Transactions on Software Engineering* 10, 728-738 (1984)

[2] Briand, L.C., Morasca, S. and Basili, V. Property-based software engineering measurement. *IEEE Transactions on Software Engineering*. 22(1). pp.68-85. (1996)

[3] Briand. L., J. Wüst, H. Lounis "A Comprehensive Investigation of Quality Factors in Object-Oriented Designs: an Industrial Case Study". *21st Int'l Conf. Software Engineering*, Los Angeles, 345-354 (1999).

[4] Calero, C., Piattini, M., Pascual, C. and Serrano, M.A. Towards Data Warehouse Quality Metrics, *Workshop on Design and Management of Data Warehouses (DMDW 20'01)*

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

[6] Genero, M., Olivas, J., Piattini, M. and Romero, F. Using metrics to predict OO information systems maintainability. *Proc. of 13th International Conference on Advanced Information Systems Engineering (CAiSE 2001).* Lecture Notes in Computer Science 2068, 388-401.

[7] ISO, ISO International Standard ISO/IEC 9126. Information technology – Software product evaluation. ISO, Geneve (2001).

[8] Jarke, M., Lenzerini, M., Vassiliou, Y. and Vassiliadis, P. *Fundamentals of Data Warehouses*, Ed. Springer (2000).

[9] Kitchenham, B., Pflegger, S., Pickard, L., Jones, P., Hoaglin, D., El-Emam, K. and Rosenberg, J.. Preliminary Guidelines for Empirical Research in Software Engineering. *IEEE Transactions of Software Engineering* 28(8) 721-734. (2002)

[10] Luján-Mora, S., Trujillo, J. and Song, I-Y. Extending UML for Multidimensional Modeling. *5th International Conference on the Unified Modeling Language* (UML 2002), LNCS 2460, 290-304.

[11] Poels G. and Dedene G.. Distance-based Software Measurement: Necessary and Sufficient Properties for Software Measures. Information and Software Technology, 42(1), 35-46. (2000)

[12] Serrano, M., Calero, C. and Piattini, M. Validating metrics for datawarehouses. *IEE Proceedings SOFTWARE*, Vol. 149, 5, 161-166 (2002)

[13] Serrano, M., Calero, C. and Piattini, M. Experimental validation of multidimensional data models metrics, *Proc of the Hawaii International Conference on System Sciences* (HICSS'36), IEEE Computer Society (2003)

[14] Trujillo, J., Palomar, M., Gómez, J. and Song, I-Y. Designing Data Warehouses with OO Conceptual Models. *IEEE Computer*, Special issue on Data Warehouses, 34 (12), 66 - 75. (2001)

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