Empirical Validation of Metrics
for Conceptual Models of Data Warehouses

Manuel Serrano[1], Coral Calero[1], Juan Trujillo[2],
Sergio Luján-Mora[2], and Mario Piattini[1]

[1] Alarcos Research Group, Escuela Superior de Informática
University of Castilla – La Mancha
Paseo de la Universidad, 4, 13071 Ciudad Real
{Manuel.Serrano,Coral.Calero,Mario.Piattini}@uclm.es
[2] Dept. de Lenguajes y Sistemas Informáticos
Universidad de Alicante
Apto. Correos 99. E-03080
{jtrujillo,slujan}@dlsi.ua.es

Abstract. Data warehouses (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 expert designers in DW's. 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: Data warehouse quality, data warehouse metrics

1 Introduction

Data warehouses, which have become the most important trend in business information technology, provide relevant and precise historical information enabling the improvement of strategic decisions [14]. A lack of quality can have disastrous consequences from both a technical and organizational points of view: loss of clients, important financial losses or discontent amongst employees [8]. Therefore, it is absolutely crucial for an organization to guarantee the quality of the information contained in these DW's from the early stages of a DW project.

The information quality of a data warehouse is determined by (i) the quality of the system itself and (ii) the quality of the data presentation (see figure 1). In fact, it is important not only that the data of the data warehouse correctly reflects the real world, but also that the data are correctly interpreted. Regarding data warehouse quality, three aspects must be considered: the quality of the DBMS (Database Manage-

ment System) that supports it, the quality of the data models[1] used in their design (conceptual, logical and physical) and the quality of the data themselves contained in the data warehouse. In this paper, we will focus on the quality of the data models, and more concretely, on the quality of conceptual models.

Regarding logical and physical models, some approaches and methodologies have been lately proposed - see [1] and [25] for a detailed classification of conceptual, logical and physical models). Even more, several recommendations exist in order to create a "good" dimensional data model - the well-known and universal star schema by [15] or [12]. However, from our point of view, we claim that design guidelines or subjective quality criteria are not enough to guarantee the quality of a "dimensional" model for DW's.
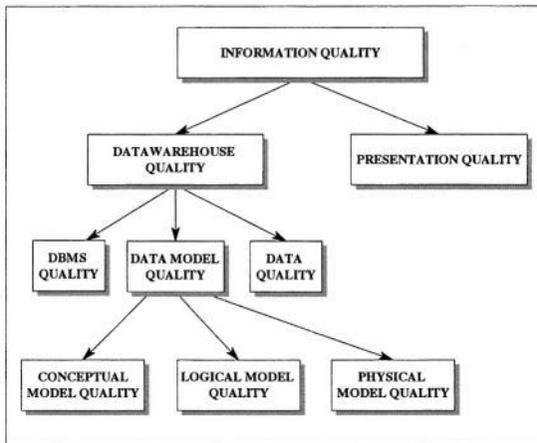

Fig. 1. Quality of the information and the data warehouse

Following this consideration, in the last years, we have been working in assuring the quality of logical models for DW's and have proposed and validated both formally [6] and empirically [20][21] several metrics that enable the evaluation of the complexity of star models (dimensional models) at the logical level.

Although conceptual modelling has not been a first priority in real world data warehouse projects, several approaches have been lately presented to represent the data warehouse information from a conceptual perspective. Some approaches propose a new notation [5][11], others use extended E/R models [19][23][7] and finally others use the UML class model [1][24][17]. Due to space constraints, we refer the reader to [1] for a detailed comparison and discussion about most of these models.

However, even using these approaches, to guarantee the quality of data warehouse conceptual models is a difficult task, with the exception of the model proposed by Jarke et al. [14], which is described in more depth in Vassiladis' Ph.D. thesis [25]. Nevertheless, even this model does not propose metrics that allow us to replace the intuitive notions of "quality" regarding the conceptual model of the data warehouse

---

[1] The term "model" refers to a modelling technique, language or model itself (eg. The E/R model) and "schema" refers to the result of applying this technique to a specific Universe of Discourse.

with formal and quantitative measures that reduce subjectivity and bias in evaluation, and guide the designer in his work. Recently, two proposals about data warehouse conceptual model metrics have appeared: [21] have proposed different metrics for model maintainability and Si-Saïd and Prat [22] have proposed some metrics for measuring multidimensional schemas analysability and simplicity. However, none of the metrics proposed so far has been empirically validated, and therefore, have not proven their practical utility. It is well known that empirical validation is crucial for the success of any software measurement project as it helps us to confirm and understand the implications of the measurement of the products [9] [16]. Thus, in this work we show a first empirical validation of the metrics proposed by [21].

The proposed metrics have been defined for guaranteeing the quality of data warehouse conceptual models, focusing on the complexity of the models, which is one of the most important factors regarding the quality in data warehouses. In defining the metrics, we have used the extension of the UML (Unified Modeling Language) presented in [24][17]. This is an object-oriented conceptual approach for data warehouses that easily represents main data warehouse properties at the conceptual level.

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 data warehouse conceptual models we will use in our study. Section 4 describes the empirical validation we have performed with the proposed metrics. Finally, Section 5 draws conclusions and introduces future investigation arising from this work.

2   Object – Oriented Conceptual Modelling with UML for Data Warehouses

In this section, we outline our approach[2] to data warehouse conceptual modelling, based on the UML. This approach has been specified by means of a UML profile[3] that contains the necessary stereotypes in order to carry out conceptual modelling successfully [17]. 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. The minimum cardinality in the role of the dimension classes is 1 to indicate that all the facts must always be related to all the dimensions. The relations "many-to-many" between a fact and a specific dimension are specified by means of the cardinality 1...* in the role of the corresponding dimen-

---

[2]   We refer the reader to (Trujillo *et al.,* 2001; Luján-Mora *et al.* 2002) for a complete description of our approach.

[3]   A *profile* is a set of improvements that extend an existing UML type of diagram for a different use. These improvements are specified by means of the extendibility mechanisms provided by UML (stereotypes, properties and restrictions) in order to be able to adapt it to a new method or model.

sion class. In our example in figure 2, we can see how the Sales fact class has a many-to-many relationship with the product dimension.

A fact is composed of measures or fact attributes. By default, all measures in the fact class are considered to be additive. For non-additive measures, additive rules are defined as constrains and are included in the fact class. Furthermore, derived measures can also be explicitly represented (indicated by /) and their derivation rules are placed between braces near the fact class.

Our approach also allows the definition of identifying attributes in the fact class (stereotype OID). In this way "degenerated dimensions" can be considered [15], thereby representing other fact features in addition to the measures for analysis. For example, we could store the ticket number (ticket_number) as degenerated dimensions, as reflected in figure 2.

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. The only prerequisite is that these classes must define a Directed Acyclic Graph (DAG) rooted in the dimension class (DAG restriction is defined in the stereotype Dimension). The DAG structure can represent both multiple and alternative path hierarchies. Every base class must also contain an identifying attribute (OID) and a descriptor attribute[4] (D). These attributes are necessary for an automatic generation process into commercial OLAP tools, as these tools store this information on their metadata.

Due to the flexibility of UML, we can also consider non-strict hierarchies (an object at a hierarchy's lower level belongs to more than one higher-level object) and complete hierarchies (all members belong to one higher-class object and that object consists of those members only). These characteristics are specified by means of the cardinality of the roles of the associations and defining the constraint {completeness} in the target associated class role respectively. See Store dimension in figure 2 for an example of all kinds of classification hierarchies. Lastly, the categorization of dimensions is considered by means of the generalization / specialization relationships of UML.

3   Metrics for Data Warehouse 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 we then 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 data warehouse schemas".

As [4] said, the structural properties (such as structural complexity) of a schema have an impact on its cognitive complexity (see figure 3). 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 [13].

---

[4]   A descriptor attribute will be used as the default label in the data analysis in OLAP tools.
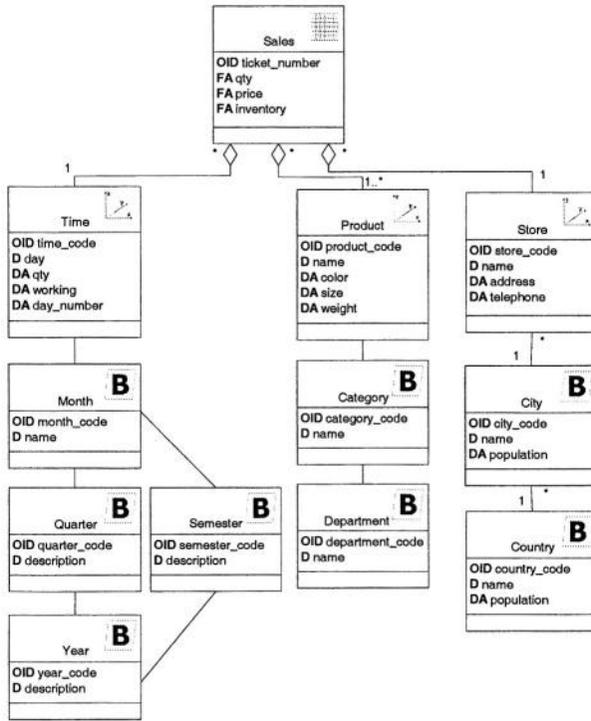
Fig. 2. Example of an Object Oriented data warehouse conceptual model using UML
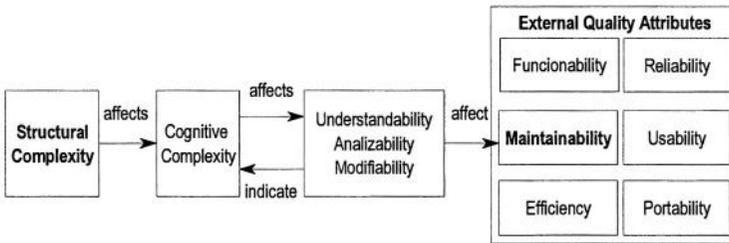
Fig. 3. Relationship between structural properties, cognitive complexity, understandability and external quality attributes, based on [4]

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

Taking into account the metrics defined for data warehouses at a logical level [20] and the metrics defined for UML class diagrams [10], we can propose an initial set of metrics for the model described in the previous section. When drawing up the proposal of metrics for data warehouse models, we must take into account 3 different levels: class, star and diagram.

In table 1 metrics for the class level are shown.

Table 1. Class scope metrics

Metric	Description
NA(C)	Number of FA (fact attributes), D (descriptor attributes) or DA (dimensional attributes) of the class C
NR(C)	Number of relationships (of any type) of the class C

The following table (see table 2) details the metrics proposed for the star level, one of the main elements of a data warehouse, composed of a fact class together with all the dimensional classes and associated base classes.

Table 2. Star scope metrics

Metric	Description
NDC(S)	Number of dimensional classes of the star S (equal to the number of aggregation relations)
NBC(S)	Number of base classes of the star S
NC (S)	Total number of classes of the star S $NC(S) = NDC(S) + NBC(S) + 1$
RBC(S)	Ratio of base classes. Number of base classes per dimensional class of the star S
NAFC(S)	Number of FA attributes of the fact class of the star S
NADC(S)	Number of D and DA attributes of the dimensional classes of the star S
NABC(S)	Number of D and DA attributes of the base classes of the star S
NA(S)	Total number of FA, D and DA attributes of the star S. $NA(S) = NAFC(S) + NADC (S) + NABC(S)$
NH(S)	Number of hierarchy relationships of the star S
DHP(S)	Maximum depth of the hierarchy relationships of the star S.
RSA(S)	Ratio of attributes of the star S. Number of attributes FA divided by the number of D and DA attributes.

Finally, in table 3, we present metrics at the diagram level of a complete data warehouse which may contain one or more stars.

Tables 4, 5 and 6 summarize the values for the defined metrics, regarding the example presented in the previous Section (figure 2). As the example has only one star, in table 6 only those values of the metrics that are different at the star and model levels are shown.

Table 3. 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.

Table 4. Class level metrics values

CLASS	NA	NR
Sales	3	3
Time	4	2
Product	4	2
Store	3	2
Month	1	3
Quarter	1	2
Semester	1	2
Year	1	2
Category	1	2
Department	1	1
City	2	2
Country	2	1

Table 5. Star level metrics values

Metric	Value
NDC(S)	3
NBC(S)	8
NC (S)	12
RBC	8/3
NAFC(S)	3
NADC(S)	11
NABC(S)	10
NA(S)	24
NH(S)	3
DHP(S)	3
RSA(S)	3/21

Table 6. Model level metrics values

Metric	Value
NFC	1
NSDC	0
NASDC	0
RDC	3

4 Empirical Validation

In this section, we will present our empirical validation for the metrics defined in the previous section. 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.

4.1 Experimental Settings

Experiment Goal Definition
The goal definition of the experiment using the GQM approximation [2] can be summarized as:

>*To analyze* the metrics for data warehouse conceptual models
>*for the purpose* of evaluating if they can be used as useful mechanisms
>*with respect of* the data warehouse maintainability
>from the designer's *point of view*
>*in the context of* practitioners

Subjects
Seventeen practitioners participated in the experiment (see table 7). The subjects work at a Spanish software consultancy that specially works on information systems development. The subjects were thirteen men and three women (one of the subjects did not give us this information), with an average age of 27.59 years. Respect to the experience of the subjects, they have an average experience of 3.65 years on computers, 2.41 years on databases, but they have little knowledge working with UML (only 0.53 years on average).

Table 7. Subjects of the experiment

Subject#	Sex	Age	Computers	Databases	UML
1	M	24	3	1	0
2	M	23	5	4	0
3	M	27	5	4	1
4	M	30	2	0	0
5	M	28	8	4	4
6	M	25	5	3	3
7	M	34	6	4	1
8	M	23	1	1	0
9	M	32	4	3	0
10	M	30	1	0	0
11	-	35	6	5	0
12	F	27	2	1	0
13	M	29	4	3	0
14	F	22	1	1	0
15	M	24	4	2	0
16	F	26	2	2	0
17	M	30	3	3	0
Mean		27,59	3,65	2,41	0,53
Minimun		22	1	0	0
Maximun		35	8	5	4
Std_Dev.		3,91	2,03	1,54	1,18

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 data warehouse schema, based on the fact that the metrics are defined in an attempt to acquire all the characteristics of a conceptual data warehouse 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 8 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. On correcting the tests we found that all the subjects answered correctly and we were therefore able to work with the results of the ten subjects.

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 8. 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 data warehouse 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 8).

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 data warehouse 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 look at the design until they were able to 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.

4.2   Collected Data Validation

After marking the test, we obtained all the times for each schema and subject (tables 9 and 10). We notice that subject 11 did not answer to the understanding tasks of schema 7 and that subjects 2 and 10 did not answer to the modification tasks on schemas 8 and 9 respectively. The times for these subjects in these exercises were considered as null values.

We decided to study the outliers before working with the average data. In order to find the outliers we made a box plot (figures 4 and 5) with the collected data (table 9 and 10). Observing these box plots (figures 4 and 5) we can observe that there are

several outliers (shown in table 11 and 13). The outliers values were eliminated from the collected data. The eliminated values are shown in tables 9 and 10 in italic font. The descriptive statistics of the final set of data can be found in tables 14 and 16. Then, we performed the analysis with this data.

Table 9. Collected data from the experiment (Understanding time)

Subject#	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10
1	60	60	30	75	128	60	35	30	96	45
2	60	89	35	120	55	85	55	75	45	48
3	45	110	30	50	45	105	40	45	70	40
4	60	30	60	90	60	60	30	60	*300*	*150*
5	65	50	30	62	50	60	30	45	40	15
6	80	55	30	240	82	85	80	45	27	27
7	125	75	30	270	70	60	60	50	45	80
8	70	64	70	180	90	90	45	45	50	30
9	65	60	50	85	100	60	65	45	65	30
10	105	82	51	89	90	48	35	101	36	38
11	60	300	120	120	120	60	-	60	180	60
12	34	55	65	115	81	111	41	72	63	54
13	45	39	48	100	48	48	75	26	70	42
14	*140*	*310*	105	90	115	*210*	105	115	*185*	*220*
15	30	12	25	34	32	43	32	28	32	18
16	90	125	90	80	90	*174*	60	110	120	110
17	57	80	30	120	150	100	82	70	38	40

Table 10. Collected data from the experiment (Modification time)

Subject#	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10
1	109	65	45	58	55	60	57	91	75	70
2	150	90	63	80	130	70	140	-	50	238
3	115	145	65	120	125	255	105	155	145	75
4	120	120	50	85	110	180	90	50	120	300
5	240	200	45	95	100	135	65	185	130	120
6	180	190	*250*	160	95	185	180	150	80	119
7	270	91	25	*355*	115	205	205	135	120	95
8	180	240	120	175	180	95	125	145	130	135
9	110	155	90	105	100	210	300	55	130	190
10	111	138	72	70	74	100	395	92	-	220
11	60	300	120	120	180	300	60	60	180	60
12	169	174	66	140	112	208	126	168	203	216
13	116	47	40	208	87	93	223	85	150	50
14	90	260	110	155	*270*	330	210	160	40	265
15	127	115	72	110	85	107	117	92	77	104
16	115	178	90	180	120	223	150	180	120	160
17	207	273	192	*330*	227	165	60	78	140	110

Validity of Results

As we know, 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.
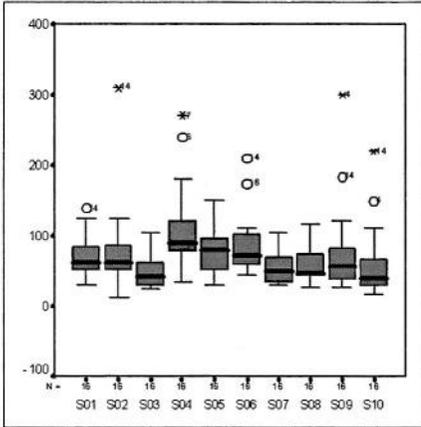
Table 11. Outliers

Schema	Subject Outliers
S01	14
S02	14
S03	
S04	6, 7
S05	
S06	14, 16
S07	
S08	
S09	4, 14
S10	4, 14

Fig. 4. Box plot of the understanding time.

Table 12. Descriptive statistics of the understanding time

	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10
Average	65,69	80,38	52,88	94,00	82,71	71,67	54,38	60,12	65,13	45,13
Minimum	30	12	25	34	32	43	30	26	27	15
Maximum	125	300	120	180	150	111	105	115	180	110
Deviation	24,79	65,04	28,90	35,01	32,78	22,22	22,36	27,36	40,55	24,31

Table 13. Outliers

Schema	Subject Outliers
S01	
S02	
S03	6
S04	7, 17
S05	14
S06	
S07	
S08	
S09	
S10	
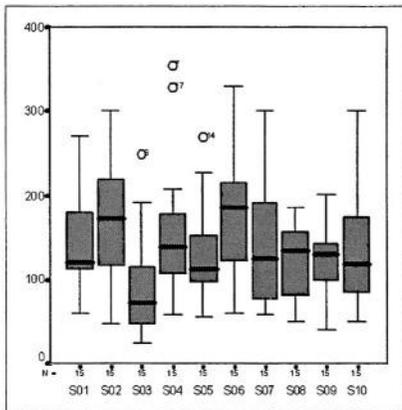
Fig. 5. Box plot of the modification time.

Table 14. Descriptive statistics of the modification time

	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10
Average	145,24	163,59	79,06	124,07	118,44	171,82	153,41	117,56	118,13	148,65
Minimum	60	47	25	58	55	60	57	50	40	50
Maximum	270	300	192	208	227	330	395	185	203	300
Deviation	55,37	73,90	41,35	44,29	43,97	79,39	91,95	46,71	44,30	76,38

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 [18] 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.
- **Differences among schemas.** The domain of the schemas were different and this could influence the results obtained in some way.
- **Precision in the time values.** The subjects were responsible for recording the start and finish times of each test. We 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 randomisation 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 experiment was 33 minutes varying from a minimum of approximately 16 minutes and a maximum of about 61 minutes. With this range of times we believe that fatigue effects hardly exist at all. Also, the different order of the tests helped to 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. The subjects wanted to participate in the experiment and to contribute to this field. 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.** Although this experiment was run by practitioners, we are aware that the number of subjects (17) could be insufficient for generalise the results. More experiments with practitioners and professionals must be carried out in order to be able to generalise the 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 [3]. We will try to obtain bigger sample data through more experimentation.

4.3 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 because we had a very small group of subjects), we decided to apply a non-parametric correlational 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 15 shows the results obtained for the correlation between each of the metrics and the tune each subject used (on each schema) to perform the task of understanding. Table 16 shows the same data for the modification tasks.

Table 15. Results of the experiment (understanding time)

Metric	NDC	NBC	NC	RBC	NAFC	NADC	NABC	NA	NH	DHP	RSA
Correlation	0,601	0,890	0,860	0,772	0,258	0,006	0,805	0,855	0,755	0,764	-0,285
Significance	0,066	0,001	0,001	0,009	0,472	0,987	0,005	0,002	0,012	0,010	0,425

Table 16. Results of the experiment (modification time)

Metric	NDC	NBC	NC	RBC	NAFC	NADC	NABC	NA	NH	DHP	RSA
Correlation	0,452	0,313	0,329	0,267	-0,459	-0,288	0,262	0,139	0,479	0,334	-0,588
Significance	0,190	0,379	0,353	0,455	0,182	0,419	0,464	0,701	0,162	0,346	0,074

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 $\alpha = 0.05$). The other metrics do not seem to be correlated with the time. On the other hand, there is not correlation at all, between the modification time and the metrics.

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 the metrics, perhaps because the modification tasks could be solved focusing only on a small part of the schema.

5   Conclusions and Future Research

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, logi-

cal and physical) used in designing them and one way of assuring the quality is using metrics.

In this paper we have focused on the empirical validation of the metrics proposed for conceptual data warehouse models as quality indicators, presenting the first experiment we have accomplished. As a result of this first experiment it seems that there exist correlation between several of the metrics and the understandability of the conceptual data warehouse models.

We are currently involved on the empirical validation of the proposed metrics process. As a result of this process the proposed metrics will be accepted, discarded or refined. When the process will finish, we will we have a set of metrics that could be used as quality indicators. These metrics could be used by the designers on their task. For example using the provided metrics they could choose among different design alternatives semantically equivalents the most maintainable one. 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 data warehousess in order to simplify their design.

Acknowledgements

References

1. Abelló, A., Samos, J. and Saltor, F. YAM2 (Yet Another Multidimensional Model): An extension of UML. *International Database Engineering & Applications Symposium (IDEAS'02)* July, pp. 172-181. (2002).
2. Basili V. and Weiss D. A Methodology for Collecting Valid Software Engineering Data, *IEEE Transactions on Software Engineering* 10, 728-738 (1984).
3. Briand L., El Emam K., Morasca S. Theoretical and empirical validation of software product measures. *Technical Report ISERN-95-03,* International Software Engineering Research Network. (1995).
4. 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).
5. Cabbibo, L. and Torlone, R. A logical approach to multidimensional databases. *Sixth International Conference on Extending Database Technology (EDBT'98),* Valencia. Spain *Lecture Notes in Computer Science 1377,* Springer-Verlag, pp 183-197. (1998).
6. Calero, C., Piattini, M., Pascual, C. and Serrano, M.A. Towards Data Warehouse Quality Metrics, *Workshop on Design and Management of Data Warehouses (DMDW'01)* (2001).

7. Cavero, J.M., Piattini, M., Marcos, E., and Sánchez, A. A Methodology for Data warehouse Design: Conceptual Modeling. *12th International Conference of the Information Resources Management Association* (IRMA2001), Toronto, Ontario, Canada. (2001).

8. English, L. *Information Quality Improvement: Principles, Methods and Management, Seminar,* 5th Ed., Brentwood, TN: Information Impact International, Inc. (1996).

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

10. 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'01).* Lecture Notes in Computer Science 2068, 388-401. (2001).

11. Golfarelli, M., Maio, D. and Rizzi, S. The Dimensional Fact Model: A Conceptual Model for Data Warehouses. International Journal of Cooperative Information Systems (IJCIS), 1998, vol. 7, 2-3, pp. 215-247. (1998).

12. Inmon, W.H. Building the Data Warehouse, third edition, John Wiley and Sons, USA, (2003).

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

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

15. Kimball, R. and Ross, M. The Data Warehouse Toolkit, second edition, John Wiley & Sons. (2002).

16. 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).

17. 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. (2002).

18. 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).

19. Sapia, C., Blaschka, M., Höfling, G. and Dinter, B. Extending the E/R Model for the Multidimensional Paradigm. *ER Workshops 1998,* Singapore, Lecture Notes in Computer Science (LNCS), vol. 1552, pp. 105-116. (1998).

20. Serrano, M., Calero, C. and Piattini, M. Validating metrics for data warehouses. *IEE Proceedings SOFTWARE,* Vol. 149, 5, 161-166. (2002).

21. 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).

22. Si-Saïd, S. and Prat, N. Multidimensional Schemas Quality : Assessing and Balancing Analyzability and Simplicity. ER 2003 Workshops. Jeusfeld, M.A. and Pastor, O. (eds.), LNCS 2814, 140-151. (2003).

23. Tryfona, N., Busborg, F. and Christiansen, G.B. starER: A Conceptual Model for Data Warehouse Design. *Proceedings of the ACM Second International Workshop on Data Warehousing and OLAP (DOLAP'99),* Kansas City, USA, pp. 3-8. (1999).

24. 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).

25. Vassiliadis, P. *Data Warehouse Modeling and Quality Issues.* Ph.D. Thesis. National Technical University of Athens. (2000).