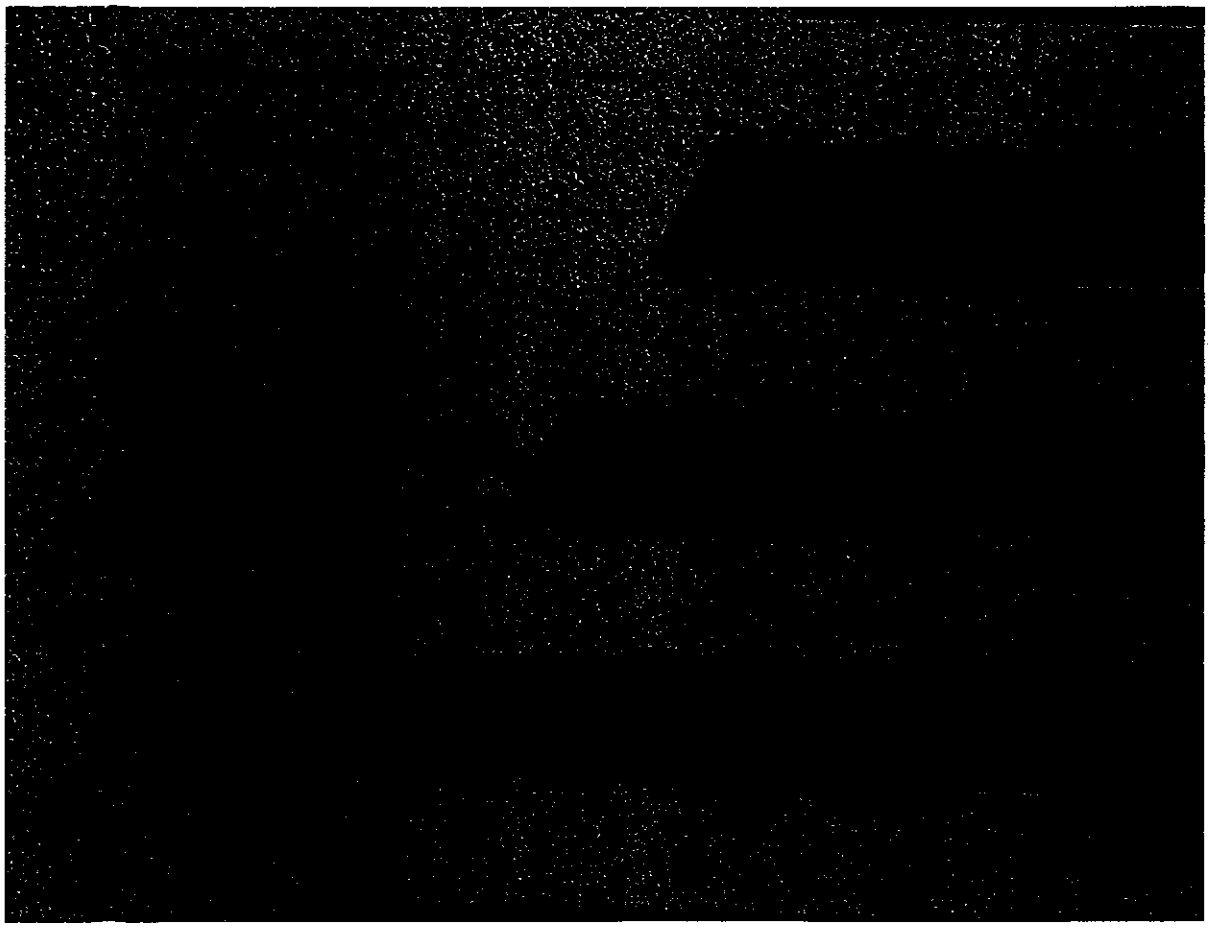
Vol. 1, No. 4
October - December 2005

An official publication of
the Information Resources
Management Association

INTERNATIONAL JOURNAL OF

DATA WAREHOUSING AND MINING



IDEA GROUP PUBLISHING
Publisher of IT books, journals and cases since 1988
www.idea-group.com

INTERNATIONAL JOURNAL OF DATA WAREHOUSING AND MINING

October - December 2005, Vol. 1, No. 4

Table of Contents

An Experimental Replication With Data Warehouse Metrics

Manuel Serrano, University of Castilla-La Mancha, Spain
Coral Calero, University of Castilla-La Mancha, Spain
Mario Piattini, University of Castilla-La Mancha, Spain

ABSTRACT

Data warehouses are large repositories that integrate data from several sources for analysis and decision support. Data warehouse quality is crucial, because a bad data warehouse design may lead to the rejection of the decision support system or may result in non-productive decisions. In the last years, we have been working on the definition and validation of software metrics in order to assure data warehouse quality. Some of the metrics are adapted directly from previous ones defined for relational databases, and others are specific for data warehouses. In this paper, we present part of the empirical work we have developed in order to know if the proposed metrics can be used as indicators of data warehouse quality. Previously, we have developed an experiment and its replication, and in this paper, we present the second replication we have made with the purpose of assessing data warehouse maintainability. As a result of the whole empirical work, we have obtained a subset of the proposed metrics that seem to be good indicators of data warehouse quality.

Keywords: *data warehouse quality; design metric; empirical validation*

INTRODUCTION

At the present time, most of the organizations face a serious problem of data pollution (Kelly, 1997), because they have great amounts of data collected at a relatively low cost that do not provide information.

Nevertheless, companies must manage the information like a product of primary importance; they must capitalize on the knowledge like a main asset; in this way, they will be able to survive and to prosper in the digital economy (Huang et al., 1999). With this aim, data warehouses arose a few years ago.

Data warehouses were created to hold data drawn from several data sources and maintained by different operating units, together with historical and summary transformations. A data warehouse is a collection of technologies aimed at enabling the knowledge worker (executive, manager, analyst) to make better and faster decisions.

Due to the increasing complexity of the data warehouses (Inmon, 1997), it is necessary to pay continuous attention to the evaluation of its quality throughout the development process. As Bouzeghoub and Kedad (2002) remark, quality in data warehouses is crucial.

A first step to obtain data warehouses with quality was the appearance of development methodologies like the proposals in Anahory and Murray (1992), Debevoise (1999), and Kimball et al. (1998).

But using development methodologies is not sufficient to assure the quality of data warehouses. Unfortunately, most of the works related to quality are focused on software quality (Arthur, 1992; Gilles, 1992; Ginac, 1998; ISO, 2001; Jones, 1997; Oskarsson & Glass, 1996), and the aspect most studied has been the program quality, disregarding the database quality (Sneed & Foshag, 1998). Even for the traditional design of databases, aspects regarding quality are not incorporated explicitly (Wang et al., 1993). All these reasons make it necessary to complement the specific methodologies with techniques, procedures, and specific metrics.

With this goal in mind, we have defined a set of metrics for measuring data warehouse star schemas in order to control its quality. Although the quality of a data warehouse depends on several factors (Jarke et al., 2000), we present in this paper the work we are developing for the logical (star) schema level.

Once the metrics have been proved as useful metrics, the data warehouse designer will be able to use them, for example, for selecting among different alternative schemas that are semantically equivalents. So, metrics could be used as design guidelines, not in the sense that they tell the designer the next step but in the sense that they can give the designer very useful information for making the best design decisions.

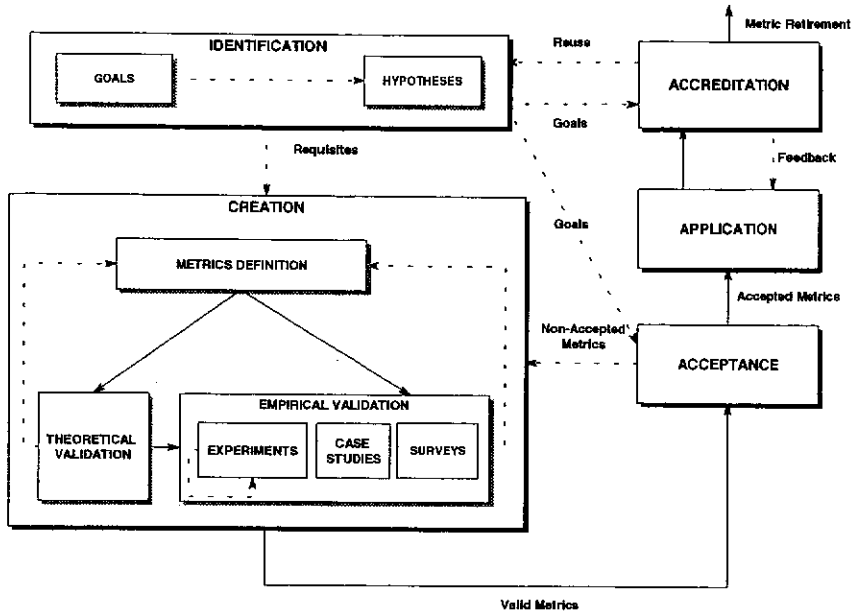
However, it is evident that when giving this powerful tool to designers, it is fundamental to ensure that metrics are really useful for the goal they are supposed to achieve. In this way, a methodological definition of the metrics is necessary to assure that metrics are useful for the goal for which they are intended. We have defined a method for defining valid and useful metrics that involves five main phases (identification, creation, acceptance, application, and accreditation). In this paper, we have focused only on the first two steps of the method (identification of requisites and metrics creation).

In the next section, we briefly present the method used to define metrics; in the third, fourth, and fifth sections, we will explain how we have used the method for the definition of data warehouse metrics. The conclusion is in the last section.

METHOD FOR DEFINING METRICS

Metric definition should be based on clear measurement goals. Metrics should be defined according to an organization's needs that are related to external quality attributes. Figure 1 presents the method we apply for obtaining correct metrics. This method has been developed using the method proposed by Calero et al. (2001b) and the MMLC (Measure Model Life

Figure 1. Metrics creation process



Cycle) by Cantone and Donzelli (2000). In this figure, continuous lines show metric flow, and dotted lines show information flow.

This method has five main phases going from the identification of goals and hypotheses to the metric application, accreditation, and retirement:

Identification. Goals of the metric are defined, and hypotheses are planned. Goals are intended to represent what we try to achieve through the measure process. Hypotheses represent the way we will develop the measure process, identifying which information must be handled in order to achieve the desired goals. This first step usually is based on expert knowledge and experience. All of the following phases will be based upon the stated goals and hypotheses, and as an outcome of this step, we obtain the requirements of the metric that we try to define.

This first phase is a very important step in getting a set of metrics, because without this phase, we cannot assure that we get clear measure goals and requisites.

Creation. The creation process of a metric is that phase in which we will create a valid metric based on the initial goals and hypotheses. As we can see in Figure 1, this step is an iterative process and has several subphases. After this phase, we get a valid metric, which should be adequate to use in real-world environments.

This is the main phase in which metric is defined and validated. This phase is divided into three subphases:

- **Metrics Definition.** Metric definition is made by taking into account the specific characteristics of the system we want to measure, the experience of the designers of these systems, and our work hypotheses. It is advisable to get

this definition in a methodological way, considering clear goals and avoiding metrics that do not meet the desired goals. A goal-oriented approach such as GQM (Goal-Question-Metric) (Basili & Weiss, 1984) also can be very useful in this step.

- **Theoretical (or Formal) Validation.** Its main goal is to probe if the intuitive idea about the measure actually is reflected in the measure process. Theoretical validations analyze the requirements that should be met and provide useful information about the mathematical and statistical operations that can be used when working with that metric. The formal validation helps us to know when and how to apply the metric. There are two main tendencies in metrics formal validation: the frameworks based on axiomatic approaches (Briand et al., 1996; Weyuker, 1988) and the ones based on measurement theory (Poels & Dedene, 2000; Whitmire, 1997; Zuse, 1998). The goal of the formers is merely definitional; on this kind of formal framework, a set of formal properties is defined for a given software attribute, and it is possible to use this property set for classifying the proposed metrics. In the case of the frameworks based on measurement theory, the information obtained is the scale to which a metric pertains, and based on this information, we can know which statistics and which transformations can be applied to the metric.
- **Empirical Validation.** The goal of this step is to prove the practical utility of the proposed metric. 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 our products. Although there are various ways to perform this step, basically we can divide the empirical validation into experiments, case studies, and surveys (Basili et al., 1999; Fenton & Pfleeger, 1997; Juristo & Moreno, 2001; Pfleeger & Kitchenham, 2001; Wohlin et al., 2000).

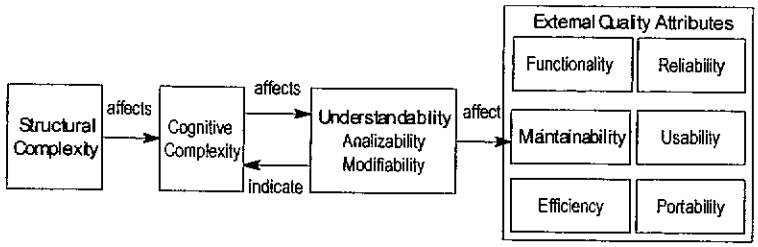
This process is evolutionary and iterative, and as a result of the feedback, the metric could be redefined or discarded, depending on its formal and empirical validation. As a result of this phase, a valid metric is obtained.

Acceptance. The aim of this phase is the systematic experimentation with the metric. This is applied in a context suitable to reproduce the characteristics of the application environment with real business cases and real users in order to verify its performance against the initial goals and stated requirements. The metric should be used in non-critical projects; in that way, the failure of a metric should not lead to crucial project failure. In order to accept a metric in this phase, we should prove that the metric goals are not affected when using it in real-world projects.

Application. The accepted metric is used in real cases. This phase runs simultaneously with the accreditation phase.

Accreditation. This is the final phase of the process. It is a dynamic phase that proceeds together with the application phase. The goal of this phase is the maintenance of the metric, so it can be adapted to application changing environment. Sometimes, the environment changes so much (i.e., changing from traditional to an object-oriented paradigm) that the metric cannot be used anymore. In those situations, metrics should be discarded, and the knowl-

Figure 2. Relationship between structural properties, cognitive complexity, understandability and external quality attributes (based on Briand et al. (1999))



edge acquired by using that metric should serve as feedback for the identification phase in order to create a new and useful metric. At the end of this phase, the metric can be retired or reused for a new metric definition process.

In the next sections, we will present all the phases applied to metrics for logical (star) schema data warehouses. In our present work, we focused only on the first two phases of the method (identification and creation), and we present them in detail.

IDENTIFICATION PHASE

As we said previously, in this phase we must specify the goals of the metrics that we plan to create, and we state the derived hypotheses. In our case, the main goal is to define a set of metrics to assess and control the quality of logical data warehouse star schemas.

Structural properties (i.e., structural complexity) of a schema have an impact on its cognitive complexity (Briand et al., 1999) (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). High cognitive complexity leads to an artefact reduction of understandability, which is conducive to undesirable external quality attributes, such as decreased maintainability (a characteristic of quality ISO 9126) (ISO, 2001).

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

CREATION PHASE

Metrics Definition

Taking into account all the information derived from the previous phase and the characteristics of a data warehouse star schema, we could define a set of metrics for star schemas (Calero et al., 2001c).

A data warehouse usually is represented as a star schema, which consists of one or more fact tables and several dimensional tables related by referential integrity relationships. The measures of interest are stored in the fact table. For each dimension of the multi-dimensional model, there exists a dimensional table that stores the information about the dimension. Based on data warehouse characteristics and the goals stated in the previous phase, we can define metrics at table, star, or schema level. We present only the metrics at

Table 1. Metrics at the schema level

NFT (Sc). Number of fact tables of the schema.
NDT (Sc). Number of dimensional tables of the schema.
NT (Sc). Number of tables of the schema. $NT(Sc) = NFT(Sc) + NDT(Sc)$
NAFT (Sc). Number of attributes of fact tables of the schema. $NAFT(Sc) = \sum_{i=1}^{NFT} NA(FT_i)$ Where NA(FTi) is the number of attributes of the fact table i of the schema Sc
NADT (Sc). Number of attributes of dimension tables of the schema. $NADT(Sc) = \sum_{i=1}^{NDT} NA(DT_i)$ Where NA(DTi) is the number of attributes of the dimensional table i of the schema Sc
NA(Sc). Number of attributes of the schema. $NA(Sc) = NAFT(Sc) + NADT(Sc)$
NFK(Sc). Number of foreign keys in all the fact tables of the schema. $NFK(Sc) = \sum_{i=1}^{NFT} NFK(FT_i)$ Where NFK(FTi) is the number of foreign keys of the fact table i of the schema Sc
RT (Sc). Ratio of dimensional tables per fact table. $RT(Sc) = \frac{NDT(Sc)}{NFT(Sc)}$
RFK(Sc). Ratio of foreign keys. (Ratio of attributes that are foreign keys). $RFK(Sc) = \frac{NFK(Sc)}{NA(Sc)}$

schema level (Table 1) because the presented experiment works at this level.

In Figure 3, an example of a star schema (used in the experiment) is presented, in which we have two fact tables (Orders and Deliveries) and four dimensional tables (Product, Date, Client, and Salesman).

The values of the metrics related to the example shown in Figure 3 are presented in Table 2.

Theoretical Validation of the Metrics

In this section, we will present the metrics theoretical validation made using the formal framework proposed by Zuse (1998). This framework is a measurement-theory-based framework, so its goal is to determine the scale to which a metric pertains. We will only show the complete process of formalization in this framework of one of the proposed metrics (NFK). The rest of the validation is made in a similar way, and the results obtained for all the metrics proposed will be presented in Table 4.

The formal framework of Zuse (1998) works with three main mathematical structures; depending on which one of these structures a metric accomplishes, we will be able to characterize it in a scale. These

Figure 3. Example of a star schema

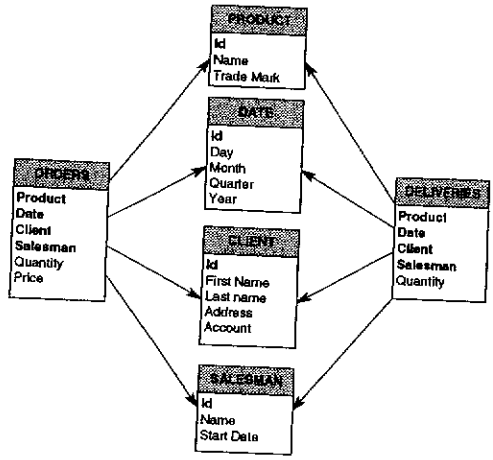


Table 2. Metrics values

Metric	Value
NFT	2
NDT	4
NT	6
NAFT	11
NADT	16
NA	27
NFK	8
RT	4/2
RFK	8/27

three structures (Table 3) are the extensive structure, the independence conditions, and the modified relation of belief. All of the details about these three structures and the complete formal framework can be found in Zuse (1998).

When a measure accomplishes the extensive structure, it also accomplishes the independence conditions and can be used on the ratio scale levels.

If a measure does not satisfy the modified extensive structure, the combination rule, which clearly describes the properties of the software measure, will exist or not, depending on the independence conditions. When a measure assumes the independence conditions but not the modified extensive structure, the scale type is the ordinal scale.

When a metric accomplishes neither the extensive structure nor the independence conditions but accomplishes the modified relation of belief, it can be characterized as *above* the level of the ordinal scale (the characterization of measures above the ordinal scale level is very important, because we cannot do very much with ordinal numbers).

NFK Metric Theoretical Validation

The NFK measure is a mapping: NFK: T -> (such that the following holds

Table 3. Summary of the mathematical structures of the Zuse's formal framework

MODIFIED EXTENSIVE STRUCTURE
Axiom1: (A, (>=) (weak order) Axiom2: A1 o A2 (>= A1 (positivity) Axiom3: A1 o (A2 o A3) ((A1 o A2) o A3 (weak associativity) Axiom4: A1 o A2 (A2 o A1 (weak commutativity) Axiom5: A1 (>= A2 (A1 o A (>= A2 o A (weak monotonicity) Axiom6: If A3 (> A4 then for any A1, A2, then there exists a natural number n, such that A1o nA3 (>A2 o nA4 (Archimedean axiom)
As we know, binary relation (>= is called weak order if it is transitive and complete: A1 (>= A2, and A2 (>= A3 (A1 (>= A3 A1 (>= A2 or A2 (>= A1
INDEPENDENCE CONDITIONS
C1: A1 (A2 (A1 o A (A2 o A and A1 (A2 (A o A1 (A o A2 C2: A1 (A2 (A1 o A (A2 o A and A1 (A2 (A o A1 (A o A2 C3: A1 (>= A2 (A1 o A (>= A2 o A, and A1 (>= A2 (A o A1 (>= A o A2 C4: A1 (>= A2 (A1 o A (>= A2 o A, and A1 (>= A2 (A o A1 (>= A o A2
Where A1 (A2 if and only if A1 (>= A2 and A2 (>= A1, and A1 (> A2 if and only if A1 (>= A2 and not (A2 (>= A1).
MODIFIED RELATION OF BELIEF
MRB1: (A, B ((: A (>= B or B (>= A (completeness) MRB2: (A, B, C ((: A (>= B and B (>= C (A (>= C (transitivity) MRB3: (A (B (A (>= B (dominance axiom) **MRB4**: $\forall (A \supset B, A \cap C = \phi) \Rightarrow (A \bullet >= B \Rightarrow A \cup C \bullet > B \cup C)$ (partial monotonicity) **MRB5**: (A ((: A (>= 0 (positivity)

for all relations between Ti and Tj (T: Ti (>= Tj (NFK(Ti) >= NFK(Tj).

In order to obtain the combination rule for NFK, we must be sure that if the concatenation (by natural join) between tables is made by foreign key, the number of foreign keys are affected (decreasing in one) and are not affected in other cases. So, we can characterize the combination rule for NFK as:

$$NFK(TioTj) = NFK(Ti) + NFK(Tj) - v$$

NFK as an Extensive Modified Structure

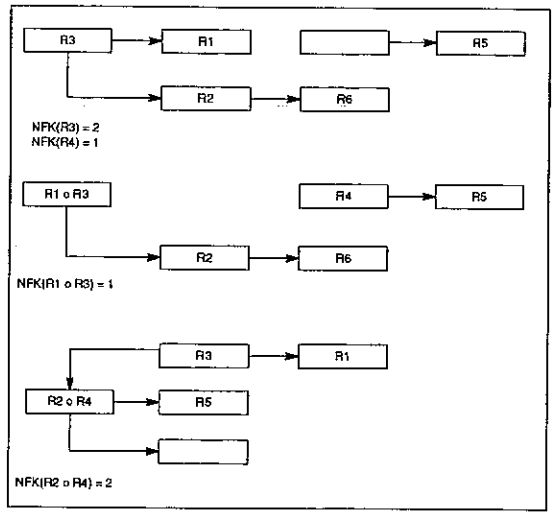
Axiom 1. T1, T2 and T3 being three tables of a schema, it is obvious that: NFK(T1) >= NFK(T2) or NFK(T2) >= NFK(T1) and also: if NFK(T1) >= NFK(T2) and NFK(T2) >= NFK(T3) (NFK(T1) >= NFK(T3). Then NFK fulfills the first axiom.

The positivity axiom (axiom 2) is not verified by the metric's own definition (when v is distinct of zero). For example,

Table 1. Theoretical validation of the metrics

	NA	NFK	NDT	NT	NADT	NAFT	NFT
Ax 1	Y	Y	Y	Y	Y	Y	Y
Ax 2	N	Y	Y	Y	Y	N	Y
Ax 3	Y	Y	Y	Y	Y	Y	Y
Ax 4	Y	Y	Y	Y	Y	Y	Y
Ax 5	N	N	N	Y	N	N	Y
Ax 6	N	N	N	Y	N	N	Y
IndC 1	N	N	N		N	N	
IndC 2	N	N	N		N	N	
IndC 3	N	N	N		N	N	
IndC 4	N	N	N		N	N	
MRB1	Y	Y	Y		Y	Y	
MRB2	Y	Y	Y		Y	Y	
MRB3	Y	Y	Y		Y	Y	
MRB4	Y	Y	Y		Y	Y	
MRB5	Y	Y	Y		Y	Y	
SCALE	AB ORD	AB ORD	AB ORD	RAT	AB ORD	AB ORD	RAT
RFK and RT pertain to the Absolute Scale							

Figure 4. Some NFK values

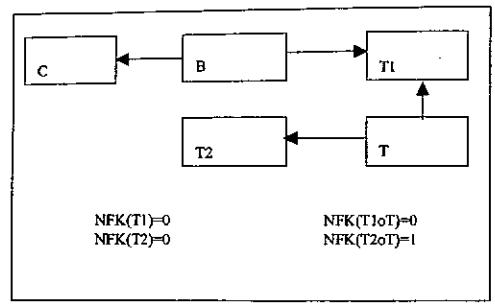


in Figure 4, we have a table T with NFK(T) = 2; however, the value of the table obtained from the concatenation of the T and the T2 tables is NFK(ToT2) = 1.

Associativity and commutativity, axioms three and four, are fulfilled, because the natural join operation is both associative and commutative.

With Figure 5, it is clear that axiom 5 may be not fulfilled, because we have that NFK(T1) = NFK(T2) but NFK(T1oT) = 0 is not greater or equal than NFK(T2oT) = 1.

Figure 5. NFK does not accomplish the Archimedean axiom



Before proving the Archimedean axiom, we must verify if the metric is idempotent; it is trivial that if a table is concatenated with itself (by natural join) more than once, the number of foreign keys increases, then the metric is not idempotent, and it is necessary to prove if NFK accomplishes the Archimedean axiom. Seeing Figure 4, we can assure that NFK does not accomplish the Archimedean axiom (NFK(R3) > NFK(R4) and NFK(R1oR3) < NFK(R2oR4)).

We can conclude that NFK is not an extensive modified structure.

NFK and the Independence Conditions

The metric does not accomplish the first condition (see Figure 4); R2 and R4 have a value equal to 1 (NFK(R2) = 1, NFK(R4) = 1). If we combine these two relations with R5, we obtain that NFK(R2oR5) = 1 and NFK(R4oR5) = 0. If the metric does not accomplish the first condition, it cannot accomplish the second one. The third condition cannot be accomplished, because the metric does not fulfill the fifth axiom of the extensive structure, and if it does not accomplish the third, it cannot accomplish the fourth. So, NFK does not accomplish the independence conditions.

NFK and the Modified Structure of Belief

Now, we must prove if NFK verifies the modified structure of belief. If the metric meets the weak order, then the first and the second axioms of the modified structure of belief are fulfilled. The third axiom is also fulfilled, because if all the foreign keys of B are included in A, then NFK(A) >= NFK(B). The weak monotonicity axiom also is accomplished, because if $A \supset B$, then NFK(A) > NFK(B); if there are not common foreign keys between A and C, neither can it be between B and C, and then the fourth conditions will be accomplished, because NFK(AoC) > NFK(BoC). The last condition, positivity, also is fulfilled, because the number of foreign keys cannot be less than zero.

In summary, we can characterize NFK as a measure above the level of the ordinal scale, assuming the modified relation of belief.

The result of the theoretical validation of the rest of metrics in the Zuse formal framework is summarized in Table 4 (where AB ORD means above the ordinal scale, and RAT means ratio scale). It is necessary to point out that, following Zuse (1998), all the metrics defined as a percentage can be characterized in the absolute scale.

As a conclusion of the theoretical validation, we have obtained that all our metrics are in the ordinal or superior scale. That means that they are theoretically valid software metrics, as remarked by Zuse (1998).

Empirical Validation of the Metrics

In this section we are going to present the empirical work we have developed with the presented metrics. As Basili et al. (1999) remark, after performing a family of experiments, it is possible to build up the cumulative knowledge to extract useful measurement conclusions to be applied in practice. So, in order to find out about the metrics, we decided to do different experiments.

Concretely, we are going to summarize the two earlier studies we developed with the metrics (Serrano et al., 2002, 2003), and after that, we will present the last experiment in depth. In all the cases, our goal is the same: trying to select which of the proposed metrics are correlated with data warehouse schema understandability. If we can conclude that some of the metrics can be used as understandability indicators, they would help data warehouse designers in the development of quality data warehouse (for example, allowing them to select the most understandable one among different semantically equivalent design alternatives). It is necessary to point out that, from our knowledge, there are not similar studies made in the data warehouse field.

Previous Experimental Work

In this section, we present the two earlier experiments we developed with the data warehouse metrics.

As a first step in the empirical process, we did an experiment (Serrano et al., 2002) for testing the correlation between the metrics and the understandability of a data warehouse schema. This experiment was performed by 12 database experts who had to rank the complexity of 11 data warehouse schemas from 1 (too easy) to 7 (too complex). The conclusion of this experiment was that it seemed that there exists a high correlation between the understandability of the schemas (through the complexity) and the metrics NFK, NFT, and NT. It was impossible to come to a satisfactory conclusion about the correlation between the metric NDT and the complexity. The other metrics did not seem to be correlated with the complexity.

Focusing only on the metrics related to the number of tables, we obtained that NFT was a good understandability indicator, whereas NDT seemed to be less correlated with understandability. So, in the second experiment (Serrano et al., 2003), we wanted to confirm the influence of the two metrics related to the number of tables (NFT and NDT) on understandability. The experiment was performed by 13 last-year students of computer science. As a result of this second experiment, we found that there was a correlation between the understandability and the NFT metric but not with the NDT metric or with the interaction between both metrics.

In Table 5[1], we summarize the results we have obtained from the first two experiments. At the end of this paper we will discuss the conclusions we can draw from the experimentation process as a whole.

Current Work

The current work was done with some of the metrics presented for logical data warehouse schema. The hypotheses we work with are the same as in the former experiments (we always want to know if there exists correlation between metrics and understandability). To describe the experiment, we use (with only minor changes) the format proposed by Wohlin, et al. (2000).

Table 5. Results summary of previous experiments

	NFK	NOT	NT	NFT	Rest of metrics
1st exp	✓	?	✓	✓	x
2nd exp	-	x	-	✓	-

Experiment Goal Definition. The goal definition of the experiment using the GQM approximation (Basili & Weiss, 1984) can be summarized as follows:

To analyze the metrics for logical data warehouse schemas
for the purpose of evaluating if they can be used as useful mechanisms
with respect of the data warehouse understandability
from the designer's *point of view*
in the context of MSc last-year students

Subjects. Eighteen students (of the last level of Computer Science MSc at the University of Castilla-La Mancha) participated in the experiment. The subjects were involved in an information retrieval course, where all the concepts related to data warehouses are discussed in depth. In addition, all of them had, in the third year of their studies, a database fundamentals course, where all the concepts related to database design and relational DBMSs were explained. So, all of them had enough knowledge on relational databases and data warehouses. We tried to define the tests involved in the experiment in such a way that they were representative of real cases.

Hypotheses Formulation. The hypotheses of our experiment are as follows:

Null hypothesis, H_0: There is no statistically significant correlation between metrics and the understandability of the schemas.

Alternative hypothesis, H_1: There is a statistically significant correlation between metrics and the understandability of the schemas.

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

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 with the metrics being researched. Table 6 presents the values for each metric in each schema.

Dependent Variables. The understandability of the tests was measured as the time each subject used to perform the task of each experimental test (which consisted of navigating among the tables of the schema in order to recover some information, similar to the SELECT sentence but in natural language). Taking into account the experience of the subjects in relational database design and the fact that the tasks were not too difficult, we thought that all of them would give correct answers. We were proved right on correcting the tests, as all the subjects were found to have an-

Table 6. Values of the metrics for the experiment schemas

Schema	NFT	NDT	NT	NACT	NADT	NA	NFK	RT	RFK
1	1	2	3	4	9	13	2	2	0,154
2	2	4	6	11	16	27	8	2	0,296
3	1	3	4	6	13	19	3	3	0,158
4	1	4	5	6	26	32	4	4	0,125
5	1	2	3	4	12	16	2	2	0,125
6	1	3	4	5	24	29	3	3	0,103

swered correctly, and therefore, we were able to work with the results of the 18 subjects. Regarding time, it is necessary to point out that this included time to analyze the schema and time to answer the questions about it.

Design. We selected a within-subject design experiment (i.e., all the tests had to be solved by each of the subjects), due to the low number of subjects. Each subject was given the schemas in a different order in order to avoid learning effects and increase internal validity.

Data Used in the Study. Six logical data warehouse schemas were used for performing the experiment. Although the domain of the schemas was different, we tried to select examples representative of real and well known cases in such a way that the results obtained were due to the difficulty of the schema, not to the complexity of the problem's domain. The documentation for each design included, in addition to the data warehouse schema, a general description and a requirements document.

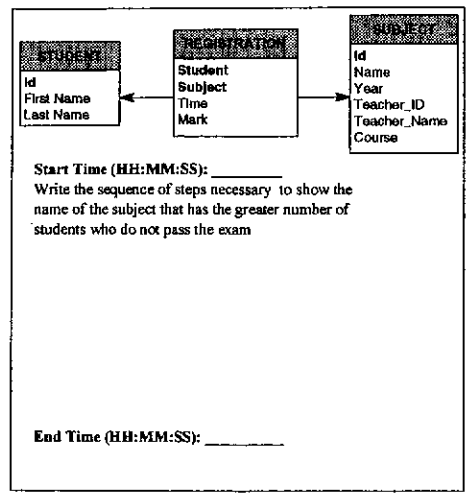
For each design, the subjects had to select some information by navigating through the tables of the schema. For example, in the schema shown in Figure 3, subjects had to answer how to obtain the name of the product with the greatest amount of items sold. In Figure 6, one of the question/answer papers is shown.

Tests were performed in less than one hour. Before starting the experiment, we explained to the subjects what the kind of exercises they had to do, the material they would be given, what kind of answers they had to provide, and how they had to record the time spent solving the problem. In this way, subjects knew that, before studying each schema, they must annotate the start time (hour, minutes, and seconds); after that, they could analyze the design and answer the given question. Once the answer to the question was written, they must annotate the final time (again in hour, minutes, and seconds). In this way, when a subject finished a test, the subject was able to go on to the next one without waiting for the rest of the experiment participants.

Tests were performed in a distinct order by different subjects in order to avoid learning effects. The way we ordered the tests was by using a randomization function (i.e., we gave a number to each schema, and the function used them for returning the order to us). To obtain the results of the experiment, we used the number of seconds needed by each subject on each schema.

Validity of Results. As we know, different threats to the validity of the re-

Figure 6. Question/answer paper



sults of an experiment exist. In this section, we will discuss threats to construct, internal, external, and conclusion validity.

Construct Validity. We propose, as a reasonable measure of analyzability, the time for determining the answer of a given question.

We must point out that the time recorded was the time used in answering the question of the test (in executing the operation) but also the time needed by the subject to analyze and to understand the initial state of the data warehouse. We have tried to assure construct validity by performing this experiment, varying the operation to be developed with respect to the previous experiments where we worked with the same hypotheses but with different operations.

Internal Validity. Regarding internal validity, the following issues should be considered:

- **Differences Among Subjects.** Within-subject experiments reduce variability among subjects. In the experiment, all the subjects had the same experience working with relational databases and data warehouses. For avoiding problems with the SQL level of knowledge, the subjects used the natural language for giving their answers.
- **Differences Among Schemas.** The domain of the schemas was different, which 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 think this method is more effective than having a supervisor who records the time of each subject. However, we are aware that subjects 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 this way, we tried to minimize learning effects.

- **Fatigue Effects.** The average time for completing the experiment was 20 minutes, varying from a minimum of approximately 14 minutes and a maximum of about 39 minutes. With this range of time, we think that fatigue effects hardly exist at all. Also, the different order of the tests helped to avoid fatigue effects.
- **Persistence Effects.** In our case, persistence effects are not present, because the subjects had never participated in a similar experiment.
- **Subject Motivation.** We told the students that the results of the experiment were similar to the ones of the exam in order to motivate them.
- **Plagiarism and Influence Among Subjects.** In order to avoid these effects, a supervisor was present during the experiment.

External Validity. Regarding external validity, the following issues should be considered:

- **Materials and Tasks Used.** We tried to use schemas and operations representative of real cases in the experiments, although more experiments with larger and more complex schemas are necessary.
- **Subjects.** Due to the difficulty of getting professionals to participate in the experiments, the experiment was done using students. We are aware that more experiments with practitioners and professionals must be carried out in order to be able to generalize the results. However, in this case, as the tasks to be performed did not require a high level of industrial experience, experiments with students could be appropriate (Basili et al., 1999). Furthermore, as stated in Hörst et al. (2000), differences between professionals and students are small, and experimentation with students is feasible under certain circumstances.

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 (six values), which perhaps is not enough for both parametric and non-parametric statistic tests (Briand et al., 1997). We are aware of this, so we will try to obtain a larger sample data through more experimentation.

For the multivariate lineal analysis, we have studied lineal model validation about the homogeneity of variance, distribution normality, and independence of residuals in order to not violate assumptions of statistical tests.

Analysis and Interpretation. We used the data collected in order to test the hypotheses formulated previously. As we cannot assure the data we collected follows a common statistical distribution (mainly because we have 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. We used a level of significance $\alpha = 0.05$.

Table 7 shows the results obtained for the correlation between each of the metrics and the time each subject used to perform the task of each experimental test.

Analyzing Table 7, we can conclude that there is a high correlation between the time used (understandability of the schemas) and the metrics NFT, NDT, NT, NAFT, NFK, and RFK (the value of significance is lower than $\alpha = 0.05$). The other metrics do not seem to be correlated with the time.

Table 7. Results of the experiment showing correlations between each metric and time

Metric	NFT	NDT	NT	NAFT	NADT	NA	NFK	RT	NFK
Correlation	0,31	0,21	0,27	0,36	0,04	0,04	0,27	-0,05	0,35
Sig.	0,00	0,03	0,01	0,00	0,71	0,71	0,01	0,62	0,00

In software engineering experimentation, multivariate analysis is commonly used, because it looks at the relationships between independent and explanatory variables but also considers the former by way of combination as covariates in a multivariate model in order to explain in a better way the variance of the dependent variables and ultimately obtain accurate predictions. We also have made a multivariate analysis using the following Multivariate Lineal Model to test our hypothesis H_0:

$$Y = \sum_{j=1}^{r} \beta_j X_j + \varepsilon$$

Where Y is the dependent variable, Xj are the independents that explain Y significantly, and ε are the residuals with N(0, σ) distribution. We use this model due to the bad results obtained using the models with intercept. Furthermore, it is reasonable to think that if all the independent variables (the metrics) are zero, then the dependent variables must be zero. We are interested in β_j (partial correlation coefficient), which means the increment of Y, if Xj is incremented in one.

The selection of the independent variables that must be included in the model can be done using different methods (e.g., backward or forward selection). The general forward selection procedure only includes the explanatory variables that are selected one at a time for inclusion in the model, so long as they fulfill certain statistical criteria. Similarly, the backward procedure starts with a model that includes all the explanatory variables, which are selected one at a time to be deleted from the model, if obey certain statistical criteria. Both procedures stop when a criterion is fulfilled (we have used 0.05 probability for inclusion and 0.10 for exclusion). After applying both methods, we have obtained the best results with the backward selection and have found that the understanding time (UT) is related to the number of fact tables (NFT), the number of dimensional tables (NDT), and the number of foreign keys (NFK):

UT = 81.333 * NFT + 0.810 * NDT + 24.595*NFK

with a p-value of 0.000 and $R^2 = 0.885$.

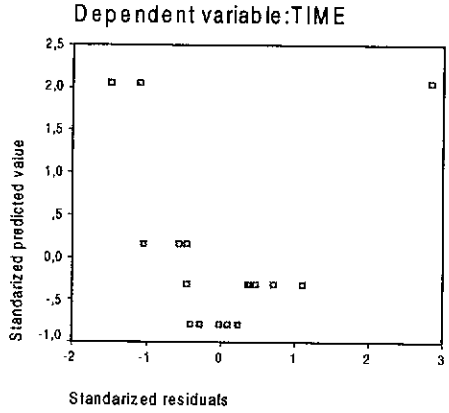
In order to not violate assumptions of statistical tests, we have studied lineal model validation about the homogeneity of variance, distribution normality, and independence of residuals.

The hypothesis to contrast the normality of residuals is: $Hnormality_0$: the standardized residual has normal distribution // $Hnormality_1$: ¬$Hnormality_0$

These hypotheses have been tested using the Kolmogorov-Smirnoff estimator (K-S), obtaining Z = 0.544 and p-value = 0.929. Looking at the p-value, we can conclude that the Understandability Time does not have normality problems[2].

The Durbin-Watson test contrasts the residual independence: $Hresiduals_0$:

Figure 7. Scatter Diagram



the residuals do not have first-order autocorrelation // $Hresiduals_i$: –$Hresiduals_0$

The obtained results were significant with respect to understandability time models at level 0.05. The residual independence does not have first-order autocorrelation.

To explore the homoscedasticity (homogeneity of variance), we looked at the scatter diagrams (Figure 7) of standardized residuals against standardized predicted values, and they did not show any deviation form.

Seeing these results, it seems that understandability is more closely related to those metrics that capture navigation in some sense among the tables. In this sense, NDT, NFT, and NT (the tables of the star) represent the nodes that it is necessary to visit for obtaining a given data (as greater is the number of tables as more complex is the selection of the concrete tables that are necessary to navigate through for obtaining a given data). The number of attributes of the fact tables may affect the understandability, as the fact table is the core of the navigation in a data warehouse. On the other hand, the number of attributes of dimensional tables may not affect the understandability, as these attributes are only consulted when their information is selected and generally are not used in the navigation. Also, it seems clear that the number of foreign keys affects the understandability of the schemas, because it represents the number of relationships the schema has, and if the number of relationships of a schema is raised, the schema becomes more complex.

The understandability is not affected by the metric ratio of tables (RT), perhaps because there are not big differences among the values of this metric in the schemas used in the experiment.

Conclusion of the Complete Experimental Work

Table 8[1] summarizes all the empirical work we have performed with the metrics for data warehouses. As a conclusion of all the experimental work, we can derive that the metrics NFT, NT, and NFK (Number of Fact Tables, Number of Tables, and Number of Foreign Keys) seem to be correlated with the understandability of data warehouse schemas.

However, with respect to NT (Number of Tables, defined as the sum of the number of fact tables plus the number of

Table 8. Experiment results summary

	NFK	NDT	NT	NFT	Rest of metrics
1st exp	✓	?	✓	✓	×
2nd exp	-	×	-	✓	-
3rd exp	✓	✓	✓	✓	-

dimensional tables), we must go on the empirical work to accept or reject the metric as an indicator of the understandability of data warehouse schemas, because it is not clear if it is a good indicator per se or it is influenced by the number of fact tables.

As we have not obtained conclusive results in the complete empirical work, we cannot confirm that NDT (Number of Dimensional Tables) is or is not an indicator of understandability of data warehouse schemas.

The results obtained for NFK are the same as the ones we obtained when we experimented with relational databases (Calero et al., 2001a).

There also could be other metrics related to dimensional tables, attributes, and foreign keys that could be correlated with the understandability of data warehouse schemas, but we must go on this empirical validation.

REMAINDER OF PHASES

We are working in collaboration with a Spanish consulting company in order to perform case studies. They also are interested in applying the obtained metrics in data warehouse development.

In this way, we look forward to going further with the method and reaching the acceptance phase, obtaining a set of good and valid metrics for data warehouse logical schemas.

CONCLUSION

Data warehouse quality is crucial, because after the data warehouse has been constructed properly, it provides the organization with a foundation that is extremely flexible and reusable (Inmon, 1997).

A first step for assuring the quality of a data warehouse design was the appearance of methodologies, but a methodology is not sufficient. We have focused our work on defining metrics for logical data warehouse schemas in order to control their quality. The definition of the metrics has been done in a methodological way.

In this paper, we have presented an experiment developed in order to find out if the presented metrics could be used as understandability indicators of a data warehouse. The experiment presented in this paper is the third we have developed with the metrics. As it is known, replication of the experiments is also necessary, because with only the isolated results of one experiment, it is difficult to appreciate how widely applicable the results are and, thus, to assess to what extent they really contribute to the field (Basili et al., 1999).

We have concluded from this experiment that there seems to be a correlation between some of the metrics (NFT, NDT, NT, NAFT, NFK, and RFK) and the understandability of the data warehouse schema.

As a conclusion from all the experimental work, we can derive that the metrics

NFT, NT, and NFK (Number of Fact Tables, Number of Tables, and Number of Foreign Keys) seem to be correlated with the understandability of data warehouse schemas. In this way, we think that this subset of metrics could be useful for designers in order to obtain more understandable data warehouse models. Using these metrics, designers can choose among several data warehouse models semantically equivalent.

As a final conclusion, we realize that the empirical work done thus far is not enough to have conclusive results. It is necessary to continue our experimental work in at least the following ways:

- Use professional subjects
- Make replications of the experiments already done (strict replications, replications that change some variables, replications that change the hypotheses, and replications that extend the theory)
- Design new experiments with more cases and different values for the metrics
- Run case studies with real data in industrial environments

After doing such empirical work, we look forward to getting a set of valid and useful metrics that can be used in real-world projects. By getting this set, we will be able to proceed to the next step of the metric definition method. We are now working with some companies in order to get a suitable environment for the acceptance phase. Also, we plan to address the remaining phases.

ACKNOWLEDGMENT

This research is part of the CALIPO project (TIC 2003-07804-C05-03) supported by the Ministerio de Ciencia y Tecnologia (Spain).

REFERENCES

Anahory, S. & Murray, D. (1997). *Data warehousing in the real world.* Harlow, UK: Addison-Wesley.

Arthur, L. (1992). *Improving software quality.* John Wiley & Sons.

Basili, V.R. & Weiss, D. (1984). A methodology for collecting valid software engineering data. *IEEE Transactions on Software Engineering, 10*(6), 728-738.

Basili, V.R., Shull, F., & Lanubille, F. (1999). Building knowledge through families of experiments. *IEEE Transactions on Software Engineering, 25*(4), 456-473.

Bouzeghoub, M. & Kedad, Z. (2002). Quality in data warehousing in information and database quality. In Piattini, Calero, & Genero (Eds.), *Information and database quality* (pp. 163-198). Kluwer Academic Publisher.

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

Briand, L., Morasca, S., & Basili, V. (1997). Response to: Comments "property-based software engineering measurement": Refining the additivity properties. *IEEE Transactions on Software Engineering, 22*(3), 196-197.

Briand. L., Wüst, J., & Lounis, H. (1999). A comprehensive investigation of quality factors in object-oriented designs: An industrial case study. In *Proceedings of the 21st International Conference on Software Engineering*, Los Angeles (pp. 345-354).

Calero, C., Piattini, M., & Genero, M. (2001a). Empirical validation of referential integrity metrics. *Information and Software Technology, 43*(15), 949-957.

Calero, C., Piattini, M., & Genero, M. (2001b). Method for obtaining correct

metrics. In *Proceedings of the Third International Conference on Enterprise and Information Systems, ICEIS'2001,* (pp. 779-784).

Calero, C., Piattini, M., Pascual, C., & Serrano, M. (2001). Towards data warehouse quality metrics. In *Proceedings of the International Workshop on Design and Management of Data Warehouses, DMDW'01*, June.

Cantone G. & Donzelli, P. (2000). Production and maintenance of software measurement models. *Journal of Software Engineering and Knowledge Engineering, 5*, 605-626.

Debevoise, N.T. (1999). *The data warehouse method.* Upper Saddle River, NJ: Prentice Hall.

Fenton, N. & Pfleeger, S. (1997). *Software metrics: A rigorous approach.* London: Chapman & Hall.

Gilles, A. (1992). *Software quality: Theory and management.* London: Chapman & Hall Computing.

Ginac, F. (1998). *Customer oriented software quality assurance.* Upper Saddle River, NJ: Prentice Hall.

Hörst, M., Regnell, B., & Wohlin, C. (2000). Using students as subjects: A comparative study of students & professionals in lead-time impact assessment. In *Proceedings of the Fourth Conference on Empirical Assessment & Evaluation in Software Engineering, EASE*, UK.

Huang, K-T., Lee, Y.W., & Wang, R.Y. (1999). *Quality information and knowledge.* Upper Saddle River, NJ: Prentice Hall.

Inmon, W.H. (2002). *Building the data warehouse* (3rd ed.). Wiley.

ISO. (2001). Software product evaluation: Quality characteristics and guidelines for their use. Geneva: ISO/IEC Standard 9126.

Jarke, M., Lenzerini, M., Vassilou, Y., & Vassiliadis, P. (2000). *Fundamentals of data warehouses.* Springer.

Jones, C. (1997). *Software quality. Analysis and guidelines for success.* Boston: International Thomson Computer Press.

Juristo N. & Moreno, A. (2001). *Basics of software engineering experimentation.* Kluwer Academic Publishers.

Kelly, S. (1997). *Data warehousing in action.* John Wiley & Sons.

Kimball, R., Reeves, L., Ross, M., & Thornthwaite, W. (1998). *The data warehouse lifecycle toolkit.* John Wiley and Sons.

Oskarsson, Ö. & Glass, R. (1996). *An ISO 9000 approach to building quality software.* Upper Saddle River, NJ: Prentice Hall.

Pfleeger, S.A. & Kitchenham, B.A. (2001). Principles of survey research. *Software Engineering Notes, 26*(6), 16-18.

Poels, G. & Dedene, G. (2000). Distance-based software measurement: Necessary and sufficient properties for software measures. *Information and Software Technology, 42*(1), 35-46.

Serrano, M., Calero, C., & Piattini, M. (2002). Validating metrics for data warehouses. In *Proceedings of the Conference on Empirical Assessment in Software Engineering, EASE 2002)*, Keele, UK, April 8-10.

Serrano, M., Calero, C., & Piattini, M. (2003). Experimental validation of multidimensional data models metrics. In *Proceedings of the Hawaii International Conference on System Sciences, HICSS'36,* January 6-9. IEEE Computer Society.

Sneed, H.M. & Foshag, O. (1998). Measuring legacy database structures. In Coombes, Hooft, & Peeters (Eds.), In *Proceedings of the European Software Measurement Conference,*

FESMA'98, Antwerp, Belgium, May 6-8 (pp. 199-210).

Wang, R.Y., Kon, H.B., & Madnick, S.E. (1993). Data quality requirements analysis and modeling. In *Proceedings of the 9th International Conference on Data Engineering*, Vienna (pp. 670-677). IEEE Computer Society.

Weyuker, E.J. (1998). Evaluating software complexity measures. *IEEE Transactions on Software Engineering, 14*(9), 1357-1365.

Whitmire, S.A. (1997). *Object oriented design measurement*. Wiley.

Wohlin, C. et al. (2000). *Experimentation in software engineering: An introduction*. Kluwer Academic Publishers.

Zuse, H (1998). A framework of software measurement. Walter de Gruyter.

ENDNOTES

1 ✓ means that there is a relationship between understandability and the metric and × means that do not exist such relationship.

2 The usual significance level in this kind of hypothesis is about 0,15 because it permits improvement of the test power.

Manuel Serrano (Manuel.Serrano@uclm.es) has an MSc and a PhD in computer science from the University of Castilla-La Mancha. He is an assistant professor at the Escuela Superior de Informática of the Castilla-La Mancha University in Ciudad Real. He is a member of the Alarcos Research Group at the same university, specializing in information systems, databases, and software engineering. His research interests are data warehouses quality and metrics and software quality.

Coral Calero (Coral.Calero@uclm.es) has an MSc and a PhD in computer science. She is an associate professor at the Escuela Superior de Informática of the Castilla-La Mancha University in Ciudad Real. She is a member of the Alarcos Research Group at the same university, specializing in information systems, databases, and software engineering. Her research interests are advanced databases design, database/data warehouse quality, Web/portal quality, software metrics, and empirical software engineering. She is the author of articles and papers in national and international conferences on these subjects.

Mario Piattini (Mario.Piattini@uclm.es) has an MSc and a PhD in computer science from the Polytechnic University of Madrid. He is the Certified Information System Auditor at ISACA (Information System Audit and Control Association) and full professor at the Escuela Superior de Informática of the Castilla-La Mancha University. He is the author of several books and papers on databases, software engineering, and information systems. He leads the ALARCOS research group of the Department of Computer Science at the University of Castilla-La Mancha in Ciudad Real, Spain. His research interests are advanced database design, database quality, software metrics, object oriented metrics, and software maintenance.