

# Metrics for data warehouse conceptual models understandability

Manuel Serrano <sup>a,\*</sup>, Juan Trujillo <sup>b</sup>, Coral Calero <sup>a</sup>, Mario Piattini <sup>a</sup>

<sup>a</sup> *Alarcos Research Group, Escuela Superior de Informática, University of Castilla – La Mancha, Paseo de la Universidad, 4 13071 Ciudad Real, Spain*

<sup>b</sup> *Dept. de Lenguajes y Sistemas Informáticos, Universidad de Alicante, Apto. Correos 99. E-03080, Spain*

Received 28 March 2006; received in revised form 5 September 2006; accepted 27 September 2006

Available online 21 November 2006

## Abstract

Due to the principal role of Data warehouses (DW) in making strategy decisions, data warehouse quality is crucial for organizations. Therefore, we should use methods, models, techniques and tools to help us in designing and maintaining high quality DWs. In the last years, there have been several approaches to design DWs from the conceptual, logical and physical perspectives. However, from our point of view, none of them provides a set of empirically validated metrics (objective indicators) to help the designer in accomplishing an outstanding model that guarantees the quality of the DW. In this paper, we firstly summarise the set of metrics we have defined to measure the understandability (a quality subcharacteristic) of conceptual models for DWs, and present their theoretical validation to assure their correct definition. Then, we focus on deeply describing the empirical validation process we have carried out through a family of experiments performed by students, professionals and experts in DWs. This family of experiments is a very important aspect in the process of validating metrics as it is widely accepted that only 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. Our whole empirical process showed us that several of the proposed metrics seems to be practical indicators of the understandability of conceptual models for DWs.

© 2006 Elsevier B.V. All rights reserved.

*Keywords:* Data warehouse quality; Data warehouse metrics; Metric validation; Data warehouse conceptual modelling

## 1. Introduction

Data warehouses (DW), which are the core of most of the current decision support systems, provide companies with many years of historical information for the decision making process [32]. A lack of quality in the data warehouse can have disastrous consequences from both technical and organizational points of view: loss of clients, important financial losses or discontent amongst employees [16]. Therefore, it is crucial for an organization to guarantee the quality of the information stored in its DW from the early stages of a DW project.

When dealing with data warehouse information quality, we have to consider different types of issues (see Fig. 1):

presentation quality and data warehouse quality. Data warehouse quality can be influenced by database management systems quality, data quality and data model quality (which can be considered at different levels, conceptual, logical and physical). Thus, one of the main issues that influence the data warehouse quality lays on the data models (conceptual, logical and physical; see Fig. 1) we use to design them. In this paper, we will focus on the quality of conceptual models as we believe that the sooner we deal with aspects regarding the data warehouse quality, we will have more chances in implementing a high quality data warehouse [53]. Our current focus is on assessing and enhancing the understandability of the data warehouse conceptual models, because as we can see on Fig. 2, understandability (among other characteristics) affects the quality of the data warehouse models.

There are several criteria for selecting the best dimensional model (e.g., understandability, maintainability, coupling, cohesion, etc.); some of them could be in conflict

\* Corresponding author. Tel.: +34 926 29 53 00; fax: +34 926 29 53 54.

E-mail addresses: [Manuel.Serrano@uclm.es](mailto:Manuel.Serrano@uclm.es) (M. Serrano), [jtrujillo@dlsi.ua.es](mailto:jtrujillo@dlsi.ua.es) (J. Trujillo), [Coral.Calero@uclm.es](mailto:Coral.Calero@uclm.es) (C. Calero), [Mario.Piattini@uclm.es](mailto:Mario.Piattini@uclm.es) (M. Piattini).

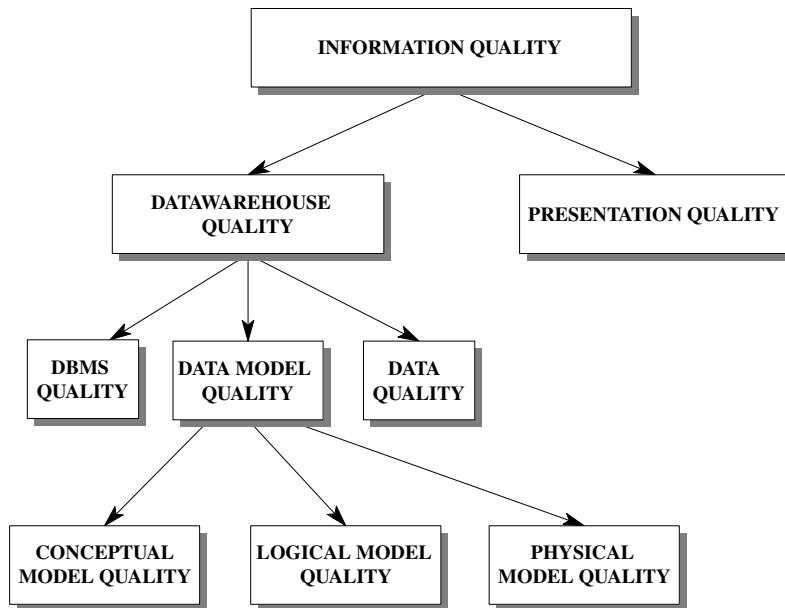


Fig. 1. Data warehouse quality.

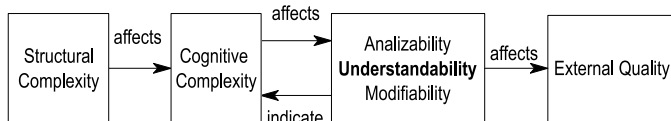


Fig. 2. Relationship between structural properties, cognitive complexity, understandability and external quality attributes.

with others. Designers should prioritize these criteria and decide which one this criteria is the most important in their work and use the metrics that fits their necessities.

Multidimensional (MD) modelling has been widely accepted as the foundation of data modelling for data warehouses. Respect to logical and physical models, some approaches and methodologies have been lately proposed – see [62]. Even more, there are several recommendations for creating “good” multidimensional data models – the well-known and universal star schema by Kimball and Ross [36] or the proposal from Inmon [27]. Nevertheless, from our point of view, we claim that design guidelines or subjective quality criteria are not enough to guarantee the quality of a data warehouse model.

The first design steps accomplished in data warehouses involve producing a conceptual schema by using a conceptual model that conveniently represents the multidimensional modelling properties. Several approaches have been lately presented to represent the multidimensional modelling properties from a conceptual perspective (see Section 2 for a more detailed list). However, none of these models tackle the quality of conceptual models for data warehouses neither with subjective nor objective (metrics) indicators. As a consequence, we may face up with several conceptual schemas for the same DW with no objective criteria that helps us decide which is the best one.

We definitely think that we need objective metrics for this purpose. It may look obvious which is the best alternative option, but intuition is not a good counsellor, we have to prove that intuitive ideas are practically valid. Metrics should be useful in supporting decisions basing on objective numbers. This objective metrics are even more important when differences between alternative schemata are not obvious. Therefore, we believe that a set of formal and quantitative measures should be provided to reduce subjectivity and bias in evaluation, and guide the designer in his work.

Getting a set of valid and useful metrics is not only a matter of *definition*; instead, it involves a complete process. This process includes, among other steps, *theoretical* and *empirical* validation of the metrics to assure the utility of the proposed metrics [17,37]. Following this consideration, we have previously *defined* a set of metrics for the conceptual modelling of data warehouses [55]. The proposed metrics have been defined for measuring the understandability of data warehouse conceptual models, focusing on the complexity of the models. In defining metrics, we have used the extension of the UML (Unified Modelling Language) presented in [59,42]. This is an object-oriented conceptual approach for data warehouses that easily represents main data warehouse properties at the conceptual level. Then, we have *theoretically* validated them in [53] using the Briand et al. [8] framework, and, in this paper, we present the theoretical validation of the proposed metrics following the DISTANCE framework [49]. Currently, we are getting involved in the empirical validation of these metrics. In [55,56], we presented the first experiments we accomplished for the empirical validation of our proposed metrics. Nevertheless, it is widely accepted that only after performing a family of experiments, it is possible to build up the cumu-

lative knowledge to extract useful measurement conclusions to be applied in practice [4].

Therefore, in this paper, we summarise the set of metrics we defined for data warehouse conceptual models and we provide their formal validation to assure their correctness. Moreover, we deeply describe the empirical validation process we have carried out through a family of experiments performed by students, professionals and experts in DWs, which highly complements our first experiment. Our family of experiments showed us that several of the proposed metrics seems to be practical indicators of the understandability of conceptual models for data warehouses.

The remainder of this paper is structured as follows: Section 2 summarises the most relevant related work. Section 3 presents the global method we follow for defining and obtaining correct metrics. Section 4 presents the identification phase of this method in which we provide the goals of our metrics. Section 5 describes the creation phase, including metric definition and a summary of the UML-based model we use for the conceptual modelling of data warehouses. Section 5.2 summarizes the theoretical validation of the proposed metrics. Section 5.3 deeply describes the family of experiments we have carried out for the empirical validation of the metrics. Finally, Section 6 draws conclusions and sketches immediate future works arising from the conclusions reached in this work.

## 2. Related work

In this section, we will organize the related work regarding the three main research topics covered by this paper: (i) multidimensional modelling, (ii) Quality issues and metrics for Software Systems in general, and (iii) quality aspects and metrics specially proposed for data warehouses.

### 2.1. Multidimensional modelling

Lately, several MD data models have been proposed. Some of them fall into the logical level (such as the well-known star-schema by R. Kimball [36]. Others may be considered as formal models as they provide a formalism to consider main MD properties. A review of the most relevant logical and formal models can be found in [6] and [1].

In this section, we will only make brief reference to the most relevant models that we consider “pure” conceptual MD models. These models provide a high level of abstraction for the main MD modelling properties at the conceptual level and are totally independent from implementation issues. One outstanding feature provided by these models is that they provide a set of graphical notations (such as the classical and well-known EER model) that facilitates their use and reading. These are as follows: *The Dimensional-Fact (DF) Model* by Golfarelli et al. [21,22], *The MultidimensionallER (M/ER) Model* by Sapia et al. [50,51], *The starER Model* by Tryfona et al. [60], the Model proposed by Hüseman et al. [25], and *The Yet Another Multidimensional Model (YAM<sup>2</sup>)* by Abelló et al. [2]. Unfortunately,

none of them has been accepted as a standard for the conceptual modelling of Data Warehouses. Recently, another approach [42,59] has been proposed as an object-oriented (OO) conceptual MD modelling approach. This proposal is a profile of the Unified Modelling Language (UML) [47], which use the standard extension mechanisms (stereotypes, tagged values and constraints) provided by the UML.

However, none of these approaches for MD modelling considers the quality of conceptual schemas as an important issue of their models and they do not neither subjective nor objective (metrics) indicators.

### 2.2. Quality issues and metrics for software systems

Software measurement is fundamental in organizations who want to reach high levels of maturity in their software processes. This fact is evidenced by the central role that measurement has in the current standards and models for process maturity and improvement such as CMMI [52], ISO 15504 [28] and the ISO/IEC 90003 [31]. From the methodological perspective, software measurement is supported by a wide variety of proposals, with the GQM (Goal Question Metric) method [61], the PSM (Practical Software Measurement) methodology [45] and the ISO 15539 [30] and IEEE 1061–1998 [26] standards deserving special attention.

There are several approaches to measuring software systems like measure the lines of code of a system, the Software Science metrics by Halstead [24], the widely used Function Points [3] or the Cyclomatic Complexity of McCabe [44].

Regarding object-oriented systems, some works have been developed in response to the high demand of metrics for such systems. Among those, we can find the proposed by Chidamber and Kemerer [14], Brito e Abreu and Carapuça [10], Lorenz and Kidd [41] and Marchesi [43], which although are metrics for an advanced design or code, some of them can be applied to conceptual schemas, such as class diagrams. We are aware that more proposals exist, but to our knowledge these are possibly the most used at a high-level design stage.

Even though several quality frameworks for data models have been proposed, most of them lack valid quantitative measures to evaluate the quality of conceptual data models in an objective way. Regarding logical data models, there are few proposals, standing out the works from [11]. On the other hand, we have found several metrics proposals for conceptual data models, like the works of Eick [15], Gray et al. [23], Kesh [35], Moody [46], and [19].

As we can see there are not too many proposals for measuring or assessing the quality of software systems, leading this situation to a lack of interest in assessing the quality of software. Fortunately, this perspective is changing, and researchers and practitioners are becoming aware of the benefits of this issue, and, nowadays, some metrics and

indicators proposals are appearing. In defining our data warehouse metrics proposal, we have considered all of these contributions.

### 2.3. Quality issues and metrics for data warehouses

As previously presented in the introduction, few works have been presented in the area of objective indicators or metrics for data warehouses; instead most of the current proposals for DWs still delegate the quality of conceptual models in the experience of the designer.

Following this idea, in the last years, we have been working in assuring the quality of data warehouse logical models and we have proposed and validated both formally [53] and empirically [53,54] several metrics for evaluate the quality of star schemas at logical level.

From our point of view, only the model proposed by Jarke et al. [32] which is described in more depth in Vassiladis' Ph.D. thesis [62] explicitly considers the quality of conceptual models for data warehouses. Nevertheless, these approaches only consider quality as intuitive notions. In this way, it is difficult to guarantee the quality of DW conceptual models, a problem which has initially been addressed by Jeusfeld et al. [33] in the context of the DWQ project. This line of research addresses the definition of metrics that allows us to replace the intuitive notions of "quality" regarding the conceptual model of the DW with formal and quantitative measures. Sample research in this

direction includes normal forms for DW design as originally proposed in [40] and generalized in [39]. These normal forms represent a first step towards objective quality metrics for conceptual schemata.

Lately, Si-Saïd and Prat [57] have proposed some metrics for measuring multidimensional schemas analyzability and simplicity. Nevertheless, none of the metrics proposed so far has been empirically validated, and therefore, have not proven their practical utility [17].

### 3. Method for defining metrics

Metric definition should be based on clear measurement goals and metrics should be defined following organisation's needs that are related to external quality attributes. In defining metrics is also advisable to take into account the experts knowledge. Fig. 3 presents the method we apply for obtaining valid and useful metrics. This method is based on the methods proposed by [12] and the MMLC (Measure Model Life Cycle [13]). 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 metrics are defined and hypotheses are formulated. All the following phases will be based upon these goals and hypotheses.

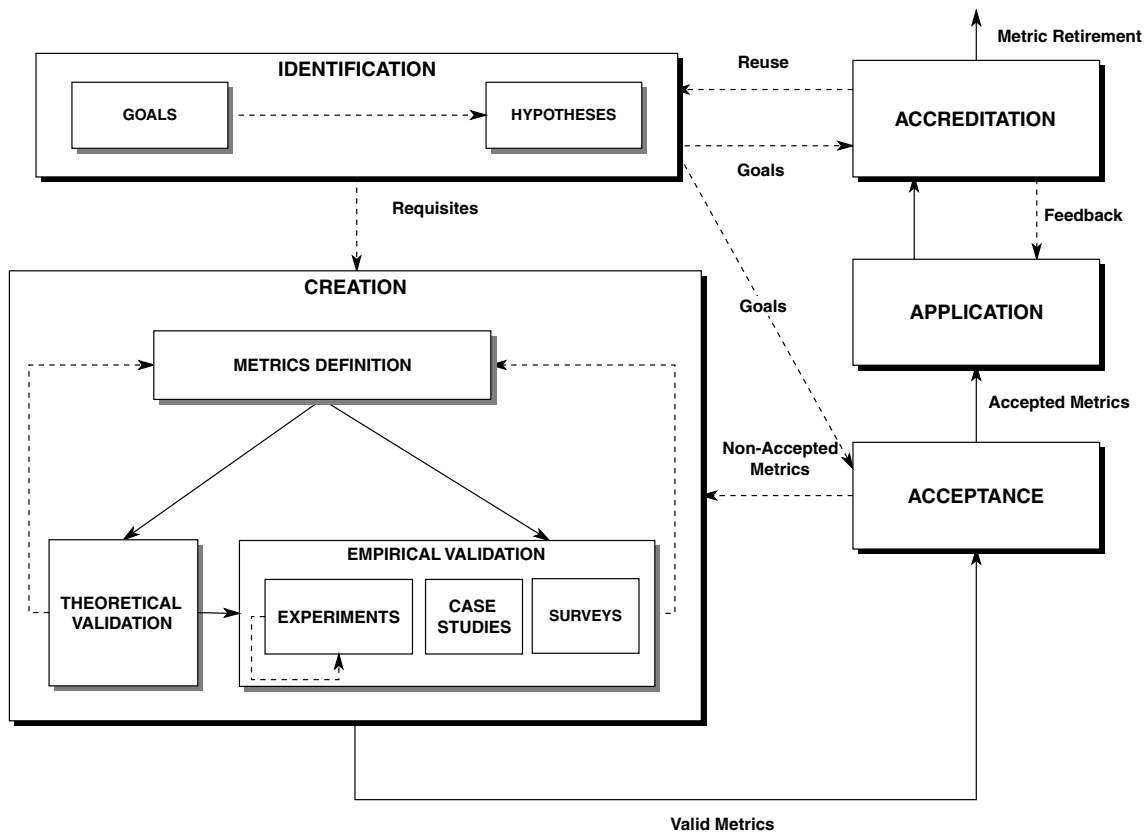


Fig. 3. Metrics creation process.

*Creation:* This is the main phase, in which metrics are defined and validated. This phase is divided into three sub phases:

*Metric definition.* Metric definition is made taking into account the specific characteristics of the system we wish to measure, the experience of the designers of these systems and our work hypotheses. A goal-oriented approach as GQM (Goal-Question-Metric [5]) can also be very useful in this step.

*Theoretical validation.* The formal (or theoretical) validation helps us to know when and how to apply the metrics. There are two main tendencies in metrics formal validation: the frameworks based on axiomatic approaches [63,8] and the ones based on measurement theory [64,66,49]. The goal of the formers is merely definitional, as on this kind of formal framework, a set of formal properties is defined for given software attributes and it is possible to use this set of properties for classifying the proposed metrics. On the other hand, in 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 of performing this step, basically, we can divide the empirical validation into: experiments, case studies and surveys [4,17,48,65,34].

This process is evolutionary and iterative and as a result of the feedback, the metric could be redefined or discarded depending on their 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 of the metric. This is applied in a context suitable to reproduce the characteristics of the application environment, with real business cases and real users, to verify its performance against the initial goals and stated requirements.

*Application:* The accepted metric is used in real cases.

*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. As a result of this phase the metric can be retired or reused for a new metric definition process. In the next sections, we will present the results of the first two phases applied to the metrics we have defined and further validated for conceptual models of data warehouses.

#### 4. Identification phase

As previously presented, in this phase we must specify the goals of the metrics 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 conceptual models of data warehouse”

Structural properties (such as structural complexity) of a model have an impact on its cognitive complexity [9] (see Fig. 2). By cognitive complexity we mean the mental burden of the persons who have to deal with the artefact (e.g., developers, testers, and maintainers). High cognitive complexity leads to an artefact reducing its analyzability, understandability and modifiability leading to reduced external quality attributes (ISO 9126 [29]).

Therefore, we can state our hypothesis as: “The proposed metrics (defined for capturing the structural complexity of conceptual models for data warehouses) can be used for controlling and assessing the quality of a data warehouse (through its understandability)”.

#### 5. Creation phase

In this section, we present the metric creation process, which involves several sub-steps as described as follows.

##### 5.1. Metric definition

Taking into account all the information derived from the previous phase and the special characteristics of the DW conceptual models that we explain in more detail in the next subsection, we can define a set of metrics for conceptual DW models.

##### 5.1.1. Object-oriented conceptual data warehouses modelling with UML

In this section, we outline our approach to conceptual modelling based on UML for the representation of structural properties of multidimensional modelling.

This approach has been specified by means of a UML profile that contains the necessary stereotypes in order to carry out conceptual modelling successfully [42]. Tables 1 and 2 summarize the defined stereotypes along with a brief description and the corresponding icon in order to facilitate their use and interpretation. These stereotypes are classified into class stereotypes (Table 1) and attribute stereotypes (Table 2). The metrics analyzed in the following sections will be performed based on this classification.

In our approach, the structural properties of multidimensional modelling are represented by means of a class diagram in which the information is organized in facts and dimensions. Some of the principal characteristics that can be represented in this model are the relationships “many-to-many” between the facts and one specific dimension, the degenerated dimensions, the multiple classification and alternative path hierarchies, and the non-strict and complete hierarchies.

Facts and dimensions are represented by means of fact classes (stereotype Fact) and dimension classes (stereotype Dimension), respectively. Fact classes are defined as compound classes in a shared aggregation relationship of  $n$  dimension classes. The minimum cardinality in the role of

Table 1  
Stereotypes of class




Name	Description	Icon
Fact	Classes of this stereotype represent facts in a MD model	
Dimension	Classes of this stereotype represent dimensions in a MD model	
Base	Classes of this stereotype represent dimension hierarchy levels in a MD model	

Table 2  
Stereotypes of attribute

Name	Description	Icon
OID	Attributes of this stereotype represent OID attributes of fact, dimension or base classes in a MD model	OID
FactAttribute	Attributes of this stereotype represent attributes of Fact classes in a MD model	FA
Descriptor	Attributes of this stereotype represent descriptor attributes of dimension or base classes in a MD model	D
DimensionAttribute	Attributes of this stereotype represent attributes of dimension or base classes in a MD model	DA

the dimension classes is  $1$  to indicate that all the facts must always be related to all the dimensions. The relationships “many-to-many” between a fact and a specific dimension are specified by means of the cardinality  $1, \dots, *$  on the role of the corresponding dimension class. A fact is composed of measurements, also called fact attributes (stereotype FactAttribute).

By default, all the measures in a fact class are considered to be additive. The semi-additive and non-additive measures are specified by means of restrictions specifying the allowed operators on certain dimensions. Furthermore, derived measures can also be represented (by means of the restriction  $/$ ) and their derivation rules are specified between brackets around the corresponding fact class. Our approach also allows the definition of identifying attributes (stereotype OID). In this way “degenerated dimensions”, which provide the facts with other characteristics in addition to the defined measures, can be represented [36].

Regarding dimensions (stereotype Dimension), each level of a classification hierarchy is represented by means of a base class (stereotype Base). An association of base classes specifies a relationship between two levels of a classification hierarchy. The only prerequisite is that these classes should define a Directed Acyclic Graph (DAG) from the dimension class (DAG restriction is defined in the stereotype Dimension). The DAG structure enables the representation of both, multiple and alternative path hierarchies. Each base class must contain an identifying attribute (stereotype OID) and a descriptive attribute<sup>1</sup> (stereotype Descriptive) in addition to the additional attributes that characterize the instances of that class.

Due to the flexibility of UML, we can consider the peculiarities of classification hierarchies as non-strict hierarchies

<sup>1</sup> The identifying attribute is used in commercial OLAP tools in order to univocally identify the instances of one hierarchy level and the descriptive attribute is the default label in the data analysis.

(an object of an inferior level belongs to more than one of a superior level) and as complete hierarchies (all the members belong to a single object of a superior class and that object is exclusively composed of those objects). These characteristics are specified by means of the role cardinality of the associations and the restriction completeness, respectively. Lastly, the categorization of dimensions is considered by means of the generalization/specialization hierarchies of UML.

In Fig. 4 we can see an example of an Object Oriented data warehouse conceptual model by using our previously described approach used in the family of experiments. In this example, we are interested in analyzing the wine sales (Fact *Wine\_sales*) of a big store. This Fact contains the specific measures to be analyzed, i.e., *qty* and *price*. On the other hand, the main dimensions along with we would like to analyze these measures are the *Time* they were sold, the specific *Wine* sold and the *Customer* to whom they were sold. Finally, Base classes *Week*, *Quarter* and *Year*; and *City* and *Country* represent the classification hierarchies of the *Time* and *Customer* dimensions, respectively, along with we are interested in analyzing measures.

### 5.1.2. Metric proposal

According to several authors [18,38], the complexity of a system is determined by the number and variety of elements and the number and variety of relationships between them. Taking into account this statement and the metrics defined for data warehouses at a logical level [54] and the metrics defined for UML class diagrams [20], 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. Class metrics refer to the attributes defined in a class (NA) and the number of relations/associations (NR) a class participates in. On the other hand, diagram metrics refer to multi-star schemas, i.e., schemas having more than one fact

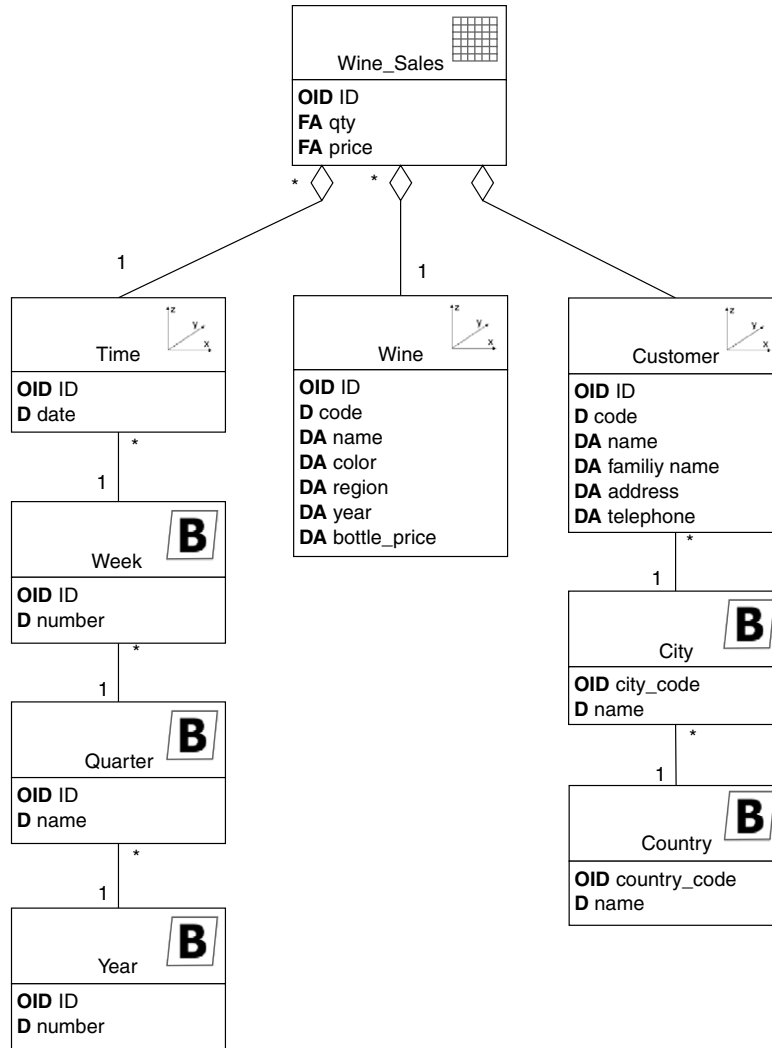


Fig. 4. Example of an object oriented data warehouse conceptual model using UML.

sharing some dimensions. Therefore, in this paper, we will focus on the star level metrics as the star schema is the main issue of a DW conceptual model.<sup>2</sup>

The following table (see Table 3) details the metrics proposed for the star level composed of a fact class together with all the dimension classes and associated base classes.

The values for the defined metrics, regarding the example presented in Section 5.1 (Fig. 4), are shown in Table 7. The example shown is the schema S09 used in the experiment.

### 5.2. Theoretical validation of the metrics

We have theoretically validated the metrics proposed using the Briand et al. framework [8], this validation can be found in [53]. In this paper, we present the proposed metrics validation using the DISTANCE framework [49].

We have chosen the DISTANCE framework because it guarantees that the metrics defined and validated using that framework are in a ratio scale.

The DISTANCE framework provides constructive procedures to model software attributes and define the corresponding measures [49]. The different procedure steps are inserted into a process model for software measurement that (i) details for each task the required inputs, underlying assumptions and expected results, (ii) prescribes the order of execution, providing for iterative feedback cycles, and (iii) embeds the measurement procedures into a typical goal-oriented measurement approach such as, for instance, GQM [5,4]. The framework is called DISTANCE as it builds upon the concepts of distance and dissimilarity (i.e., a non-physical or conceptual distance). This distance-based measure construction process consists of five steps:

- Step 1. Find a measurement abstraction
- Step 2. Model distances between measurement abstractions

<sup>2</sup> Once star level metrics are validated and accepted, the next step of our works will be validating diagram level metrics.

Table 3  
Star scope metrics

Metric	Description
NDC(S)	Number of dimension classes of the star S (equal to the number of aggregation relationships)
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 dimension 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 dimension 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

- Step 3. Quantify distances between measurement abstractions
- Step 4. Find a reference abstraction
- Step 5. Define the software measure

### 5.2.1. NDC theoretical validation

The Number of Dimension Classes (NDC) measure is defined at the diagram level as the total number of dimension classes within a data warehouse conceptual model.

In the following, we will follow each of the steps for measure construction proposed in the DISTANCE framework. In order to exemplify the process we will use the models shown in Fig. 5.

- *Step 1. Find a measurement abstraction.* In our case the set of software entities  $P$  is the Universe of data warehouse conceptual models (UDCM) that is relevant for some Universe of Discourse (UoD) and  $p$  is a Data warehouse Conceptual Model (DCM) (i.e.,  $p \in \text{UDCM}$ ). The attribute of interest *attr* is the number of dimension classes, i.e., a particular aspect of DCM

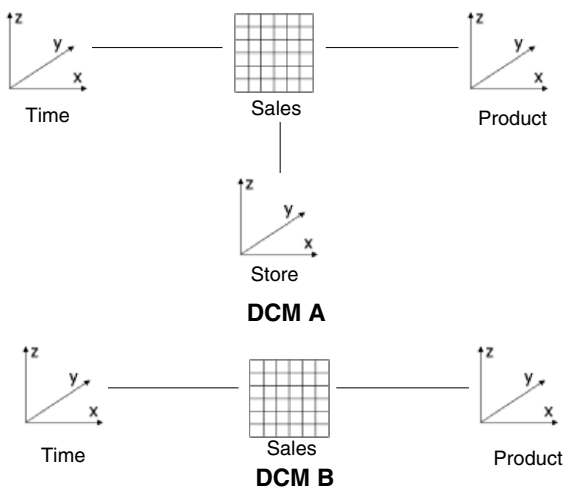


Fig. 5. Two examples of conceptual models of data warehouse.

structural complexity. Let UDC be the Universe of Dimension Classes relevant to the UoD. The set of dimension classes within a DCM, called  $\text{SDC}(\text{DCM})$  is then a subset of UDC. All the sets of dimension classes within the DCMs of UDCM are elements of the power set of UDC, denoted by  $\wp(\text{UDC})$ . As a consequence we can equate the set of measurement abstractions  $M$  to  $\wp(\text{UDC})$  and define the abstraction function as:

$$abs_{\text{NDC}} : \text{UDCM} \rightarrow \wp(\text{UDC}) : \text{DCM} \rightarrow \text{SDC}(\text{DCM})$$

This function simply maps a DCM onto its set of dimension classes. In our example we have the set of dimension classes of DCM A and of DCM B:

$$\begin{aligned} abs_{\text{NDC}}(\text{DCM A}) &= \text{SDC}(\text{DCM A}) = \{\text{Time, Store, Product}\} \\ abs_{\text{NDC}}(\text{DCM B}) &= \text{SDC}(\text{DCM B}) = \{\text{Time, Product}\} \end{aligned}$$

- *Step 2. Model distances between measurement abstractions.* The next step is to model distances between the elements of  $M$ . We need to find a set of elementary transformation types for the set of measurement abstractions  $\wp(\text{UDC})$  such that any set of dimension classes can be transformed into any other set of dimension classes by means of a finite sequence of elementary transformations. Finding such a set is quite easy in case of a power set. Since the elements of  $\wp(\text{UDC})$  are sets of dimension classes,  $T_e$  must only contain two types of elementary transformations: one for adding a dimension class to a set and one for removing a dimension class from a set. Given two sets of dimension classes  $s_1 \in \wp(\text{UDC})$  and  $s_2 \in \wp(\text{UDC})$ ,  $s_1$  can always be transformed into  $s_2$  by removing first all the dimension classes from  $s_1$  that are not in  $s_2$ , and then adding all the dimension classes to  $s_1$  that are in  $s_2$ , but were not in the original  $s_1$ . In the ‘worst case scenario’,  $s_1$  must be transformed into  $s_2$  via an empty set of attributes. Formally,  $T_e = \{t_{0\text{-NDC}}, t_{1\text{-NDC}}\}$ , where  $t_{0\text{-NDC}}$  and  $t_{1\text{-NDC}}$  are defined as:

$$t_{0\text{-NDC}} : \wp(\text{UDC}) \rightarrow \wp(\text{UDC}) : s \rightarrow s \setminus \{a\}, \text{ with } a \in \text{UDC}$$

$$t_{1\text{-NDC}} : \wp(\text{UDC}) \rightarrow \wp(\text{UDC}) : s \rightarrow s \cup \{a\}, \text{ with } a \in \text{UDC}$$



In our example, the distance between  $abs_{NDC}(\text{DCM A})$  and  $abs_{NDC}(\text{DCM B})$  can be modelled by a sequence of elementary transformations that does not remove any dimension class from  $\text{SDC}(\text{DCM A})$  and that adds Store to  $\text{SDC}(\text{DCM A})$ . This sequence of 1 elementary transformations is sufficient to transform  $\text{SDC}(\text{DCM A})$  into  $\text{SDC}(\text{DCM B})$ . Of course, other sequences exist and can be used to model the distance in sets of dimension classes between DCM A and DCM B. But it is obvious that no sequence can contain fewer than 1 elementary transformation if it is going to be used as a model of this distance. All 'shortest' sequences of elementary transformations qualify as models of distance.

- *Step 3. Quantify distances between measurement abstractions.* In this step the distances in  $\wp(\text{UDC})$  that can be modelled by applying sequences of elementary transformations of the types contained in  $T_e$ , are quantified. A function  $\delta_{NDC}$  that quantifies these distances is the metric (in the mathematical sense) that is defined by the symmetric difference model, i.e., a particular instance of the contrast model of Tversky [58]. It has been proven in [49] that “the symmetric difference model can always be used to define a metric when the set of measurement abstractions is a power set”.

$$\delta_{NA} : \wp(\text{UDC}) \times \wp(\text{UDC}) \rightarrow \mathfrak{R} : (s, s') \rightarrow |s - s'| + |s' - s|$$

This definition is equivalent to stating that the distance between two sets of dimension classes, as modelled by a shortest sequence of elementary transformations between these sets, is measured by the count of elementary transformations in the sequence. Note that for any element in  $s$  but not in  $s'$  and for any element in  $s'$  but not in  $s$ , an elementary transformation is needed. The symmetric difference model results in a value of 1 for the distance between the set of dimension classes of DCM A and DCM B. Formally,

$$\begin{aligned} \delta_{NDC}(abs_{NDC}(\text{DCM A}), abs_{NDC}(\text{DCM B})) \\ &= |\{\text{Time, Store, Product}\} - \{\text{Time, Product}\}| \\ &\quad + |\{\text{Time, Product}\} - \{\text{Time, Store, Product}\}| \\ &= |\{\text{Store}\}| + |\{\}\} = 1 \end{aligned}$$

- *Step 4. Find a reference abstraction.* In our example, the obvious reference point for measurement is the empty set of dimension classes. It is desirable that an DCM without dimension classes will have the lowest possible value for the NDC measure. So that we define the following function:

$$ref_{NDC} : \text{UDCM} \rightarrow \wp(\text{UDC}) : \text{DCM} \rightarrow \emptyset$$

- *Step 5. Define the software measure.* In our example, the number of dimension classes of a Data warehouse Conceptual Model  $\text{DCM} \in \text{UDCM}$  can be defined as the distance between its set of attributes  $\text{SDC}(\text{DCM})$  and

the empty set of dimension classes  $\emptyset$ , as modelled by any shortest sequence of elementary transformations between  $\text{SDC}(\text{DCM})$  and  $\emptyset$ . Hence, the NDC measure can be defined as a function that returns for any  $\text{DCM} \in \text{UDCM}$  the value of the metric  $\delta_{NDC}$  for the pair of sets  $\text{SDC}(\text{DCM})$  and  $\emptyset$ :

$$\begin{aligned} \forall \text{DCM} \in \text{UDCM} : \text{NDC}(\text{DCM}) &= \delta_{NDC}(\text{SDC}(\text{DCM}), \emptyset) \\ &= |\text{SDC}(\text{DCM}) - \emptyset| + |\emptyset - \text{SDC}(\text{DCM})| \\ &= |\text{SDC}(\text{DCM})| \end{aligned}$$

As a consequence, a measure that returns the count of dimension classes in a data warehouse conceptual model qualifies as a number of dimension classes measure. And this proves the validity of the NDC metric from a theoretical perspective. It must be noted here that, although this result seems trivial, other measurement theoretical approaches to software measure definition cannot be used to guarantee the ratio scale type of the NDC measure. The number of dimension classes in a DCM can, for instance, not be described by means of a modified extensive structure, as advocated in the approach of Zuse [66], which is the best known way to arrive at ratio scales in software measurement.

### 5.2.2. Other metrics validation

Due to space constraints, describing the construction process and theoretical validation for all the other proposed metrics would lead us to an extremely long paper, and therefore, we do not provide it in detail.<sup>3</sup> However, the process is analogous and is summarized in Table 4. As all the metrics have been defined following the distance-based process for metric construction, all the metrics are defined as distances. This fact guarantees that all the metrics are characterised by the ratio scale. That means that they are theoretically valid software metrics because they are in the ordinal or in a superior scale, as remarked by Zuse [66], and are therefore perfectly usable.

### 5.3. Empirical validation

In this section, we present the empirical work we have developed with the previously presented metrics. As Basili et al. [4] remarks, 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. Therefore, in order to find out about the metrics we decided to do different experiments.

Let us summarize the two previous studies developed with the metrics [55,56] and then we will deeply present the last experiment we have carried out. In all the cases our goal is the same: trying to select which of the proposed metrics are correlated with data warehouse conceptual schema understandability. If we conclude that some of

<sup>3</sup> Please, refer to [53] for a detail description of the whole process for all metrics.

Table 4  
Abstraction functions for the rest of the metrics

Metric	Abstraction function
NDC	$abs_{NDC}: UDCM \rightarrow \wp(UC): DCM \rightarrow SDC(DCM)$ where UDCM is the Universe of Data Warehouse Conceptual Models UC is the Universe of Classes relevant to an UoD $SDC(DCM) \subseteq UC$ is the set of dimension classes within a model
NBC	$abs_{NBC}: UDCM \rightarrow \wp(UC): DCM \rightarrow SBC(DCM)$ where UDCM is the Universe of Data Warehouse Conceptual Models UC is the Universe of Classes relevant to an UoD $SBC(DCM) \subseteq UC$ is the set of base classes within a model
NC	$abs_{NC}: UDCM \rightarrow \wp(UC): DCM \rightarrow SC(DCM)$ where UDCM is the Universe of Data Warehouse Conceptual Models UC is the Universe of Classes relevant to an UoD $SC(DCM) \subseteq UC$ is the set of classes within a model
NADC	$abs_{NADC}: UDCM \rightarrow \wp(UA): DCM \rightarrow SAD(DCM)$ where UDCM is the Universe of Data Warehouse Conceptual Models UA is the Universe of Attributes relevant to an UoD $SAD(DCM) \subseteq UA$ is the set of attributes of the dimension classes within a model
NAFC	$abs_{NAFC}: UDCM \rightarrow \wp(UA): DCM \rightarrow SAF(DCM)$ where UDCM is the Universe of Data Warehouse Conceptual Models UA is the Universe of Attributes relevant to an UoD $SAF(DCM) \subseteq UA$ is the set of attributes of the fact classes within a model
NABC	$abs_{NABC}: UDCM \rightarrow \wp(UA): DCM \rightarrow SAB(DCM)$ where UDCM is the Universe of Data Warehouse Conceptual Models UA is the Universe of Attributes relevant to an UoD $SAB(DCM) \subseteq UA$ is the set of attributes of the base classes within a model
NA	$abs_{NA}: UDCM \rightarrow \wp(UA): DCM \rightarrow SA(DCM)$ where UDCM is the Universe of Data Warehouse Conceptual Models UA is the Universe of Attributes relevant to an UoD $SA(DCM) \subseteq UA$ is the set of attributes within a model
NH	$abs_{NH}: UDCM \rightarrow \wp(UH): DCM \rightarrow SH(DCM)$ where UDCM is the Universe of Data Warehouse Conceptual Models UH is the Universe of generalization relationships relevant to an UoD $SH(DCM) \subseteq UH$ is the set of generalization relationships within a model
DHP	Metric DHP is defined at class level as: $abs_{DHP}: UC \rightarrow \wp(UC): C \rightarrow SLongestPath(C)$ where UC is the Universe of Classes $SLongestPath(C) \subseteq UC$ is the set of classes related by generalization relationships In case of multiple relationships, only the classes in the longest path are considered Metric DHP at model class is the maximum value of DHP calculated for all the classes of the model
RSA and RBC	These metrics cannot be defined using the DISTANCE framework, as the framework only considers lineal distances between entities and these metrics are defined as combination of several metrics. However, being defined as a function of valid metrics, these metrics can be considered valid

the metrics can be used as understandability indicators, they would help data warehouse designers in the design of quality data warehouses (for example, allowing them to select among different design alternatives semantically equivalent the most understandable one).

### 5.3.1. Previous experimental work

In this section we summarize the two previous experiments developed with the data warehouse metrics.

The first experiment [55] was performed by 17 professionals working in a Spanish software consultancy that specialized in 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).

In the second experiment [56] we replicated the first one using as experimental subjects twenty-eight last course students in MSc of Computer Science from University of Castilla – La Mancha (Spain). The subjects were twenty three men and five women, with an average age of 24.5 years. All the subjects had almost the same experience as they are all students.

In both experiments, subjects attended to an explanatory session in which we explained the basics of the data warehouses conceptual modelling and we told them how to complete the exercises they were going to face. Subjects in the experiments had to analyze 10 data warehouse conceptual models and they had to do some exercises. The experimental package can be found at <http://alarcos.inf-cr.uclm.es/english/research.html>

For analyzing the experimental data we collected the time that was spent in doing the exercises by the subjects. And we tried to find if there was any type of relationship between this understandability time and the proposed metrics.

In the first experiment, we found that there exists a high correlation between the understandability of the conceptual models and the metrics NBC, NC, RBC, NABC, NA, NH and DHP (Number of Base Classes, Number of Classes, Ratio of Base Classes, Number of Attributes of Base Classes, Number of Attributes, Number of Hierarchies and Depth of Hierarchy Path, respectively). In the second experiment we found the same results as in the first one.

In Table 5, we summarize the results obtained from the first two experiments. In that table we can see that there exists a high correlation between the metrics NBC, NC, RBC, NABC, NH and DHP and the understandability of the schemas. This lead us to think that the amount of classes and hierarchies has an impact on the understandability of conceptual data warehouse schemas. At the end of this paper, we will discuss the conclusions we can draw from the experimentation process as a whole.

### 5.3.2. Current work

In this section, we will present the current empirical validation for the defined metrics. This time we tried to corroborate the previous obtained results replicating the experiment with database and UML experts and lecturers from the University of Alicante (Spain). In this experiment, we wanted to take a step further and we tested, not only the understandability time, but also the efficiency and effectiveness of the subjects when dealing with data warehouse conceptual schemas.

In order to describe all the experimental process, we firstly define the experimental settings (including the main

goal of our experiment, the subjects who participated in the experiment, the main hypotheses under which we run out the experiment, the independent and dependent variables used in our model, the experimental design, the experiment running, the material used and the subjects that performed the experiment). Then, we will discuss about the collected data validation. Finally, we analyse and interpret the results to find out if they follow the formulated hypotheses or not.

#### 5.3.2.1. Experimental settings.

##### Experiment goal definition

The goal definition of the experiment using GQM [5] can be summarized as:

*To analyze the metrics for data warehouse conceptual models for the purpose of evaluating if they are useful with respect of the data warehouse understandability, efficiency and effectiveness. from the researcher's point of view in the context of experts*

##### Subjects

Twenty-five experts from the University of Alicante (Spain) participated in the experiment (see Table 6). All of them were lecturers in the University of Alicante (Spain). The subjects were 16 men and 8 women (one of the subjects did not give us this information), with an average age of 28.52 years. Respect to the experience of the subjects, they have an average experience of 10.08 years on computers, 5.08 years on databases and they have little knowledge working with UML (only 1.80 years on average).

##### Hypotheses formulation

The hypotheses of our experiment are:

*Null hypothesis, H<sub>01</sub>*: There is no a statistically significant correlation between the metrics and the understandability time of the data warehouse conceptual data models.

*Null hypothesis, H<sub>02</sub>*: There is no a statistically significant correlation between the metrics and the efficiency of the subjects when dealing with data warehouse conceptual data models.

*Null hypothesis, H<sub>03</sub>*: There is no a statistically significant correlation between the metrics and the effectiveness of the subjects when dealing with data warehouse conceptual data models.

Alternative hypothesis, H<sub>11</sub>:  $\neg H_{01}$

Table 5

Results summary of previous experiments (✓ means that there is a relationship between understandability and the metric)

	NDC	NBC	NC	RBC	NAFC	NADC	NABC	NA	NH	DHP	RSA
1st exp		✓	✓	✓			✓	✓	✓	✓	
2nd exp		✓	✓	✓			✓	✓	✓	✓	

Table 6  
Subjects of the experiment (data in years)

Subject#	Sex	Age	Computers	Databases	UML
1	M	25	7	6	3
2	M	29	12	8	4
3	M	25	12	5	4
4	F	25	7	5	3
5	M	37	20	0	0
6	M	29	13	7	3
7	M	26	8	2	0
8	M	24	7	5	2
9	M	36	18	0	0
10	F	24	6	3	0
11	M	38	14	12	1
12	M	35	7	3	1
13	M	22	7	4	2
14	M	30	5	0	0
15	F	35	16	9	2
16	F	30	12	8	3
17	M	37	18	10	5
18	F	28	9	9	1
19	M	34	15	3	0
20	F	24	6	4	3
21	F	25	8	6	5
22	M	24	6	5	1
23	M	23	6	5	1
24	–	24	7	4	1
25	F	24	6	4	0
Mean		28.52	10.08	5.08	1.80
Minimum		22	5	0	0
Maximum		38	20	12	5
Std_Dev.		5.24	4.54	3.09	1.63

Alternative hypothesis,  $H_{12}$ :  $\neg H_{02}$

Alternative hypothesis,  $H_{13}$ :  $\neg H_{03}$

Alternative hypotheses are stated to determine if there is any kind of interaction between the metrics and the factor we want to test, 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 this variable corresponds to the structural complexity, which is measured through the metrics being

researched. Table 7 presents the values for each metric in each DW conceptual schema provided in the experiment (see next sub-section).

*Dependent variables.* The understandability of the tests was measured as the time each subject used to perform the tasks of each experimental test. The experimental task consisted in understanding the models and answer to some questions about the models. For measuring the efficiency we use the next formula:

$$\text{Efficiency} = \frac{\text{Number of correct answers}}{\text{Time}}$$

Regarding Effectiveness, we calculated it in this way:

$$\text{Effectiveness} = \frac{\text{Number of correct answers}}{\text{Number of questions}}$$

#### Material design and experiment running

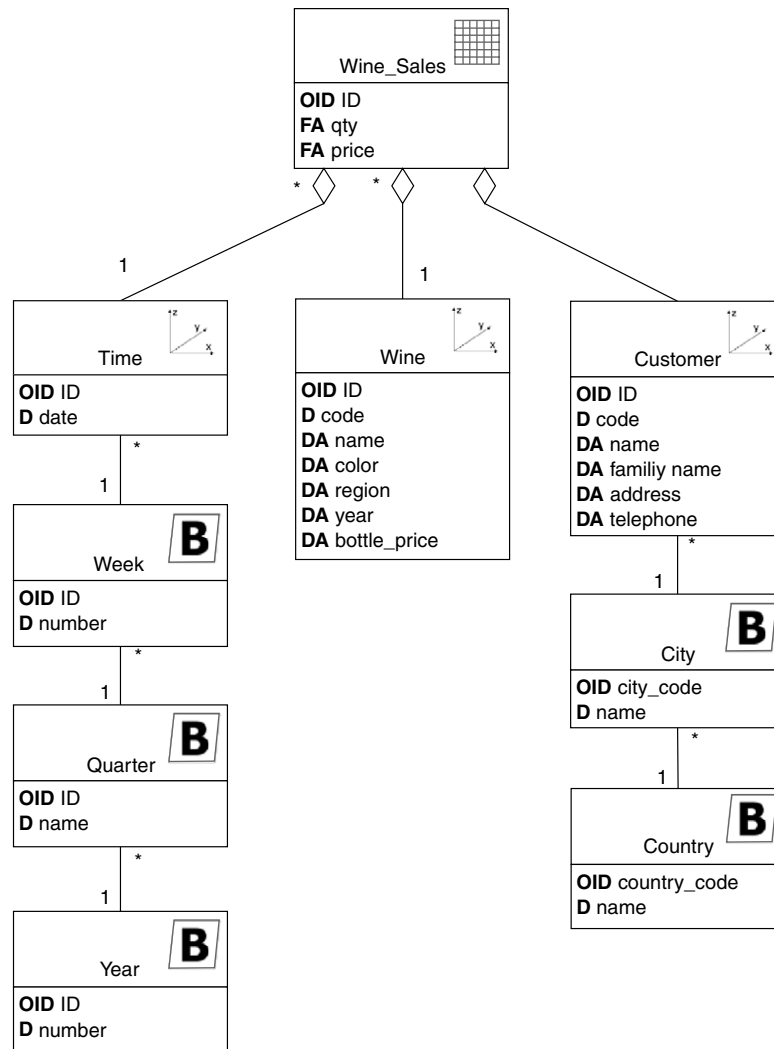
Ten conceptual data warehouse schemas were used for performing this experiment. Although the domain of the schemas was different, we tried to select representative examples of real world cases in such a way that the results obtained were due to the difficulty of the schema and not to the complexity of the domain problem. We tried to have schemas with different metrics values (see Table 7). In order to look up at the schemas, we refer the reader to <http://alarcos.inf-cr.uclm.es/english/research.html>, where the experimental packages can be found. An example of one of the sheets used in the experiment is shown in Fig. 6

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 and answer some questions about the design. The experimental tasks were constructed using our experience in working with data warehouse real cases, and therefore, we can consider these tasks significant for the examples and similar to real world tasks. Also the domains of the schemata were common and well known to avoid problems with domain understanding.

Before starting the experiment, we explained to the subjects the kind of exercises that they had to perform, the

Table 7  
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



**Write the starting time (HH:MM:SS):**

1) Answer to this questions:

1. Which classes do you need to use for knowing the color of one wine?
2. Which classes do you need to use for obtaining a list of all the sales of a year?

2) Make the necessary modifications to the model to fit this requirements:

1. You need to store information about the taxes of each sale
2. You need to store information about the month to which a week belongs to
3. You need to store information about the promotions made with the wines

**Write the finishing time (HH:MM:SS):**

Fig. 6. Example of experimental material.

material that they would be given, what kind of answers they had to provide and how they had to record the time spent performing the tasks. We also explained to them that before studying each schema they had to annotate the starting 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 had been written, they had to annotate the final time (again in hour, minutes and seconds).

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 schema by each subject. We also check the experiments for correct answers.

**5.3.2.2. Collected data validation.** Before collecting time, we marked all the tests to be sure that the provided answers were correct. When we obtained all the times for each schema and subject (Table 8a), we notice that subject 5 did not answer to the task of schema 6. We also noticed that the time spent by subject 19 in the questions of the schema 9

Table 8  
Collected data from the experiment

Subject#	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10
<i>(a) Understanding time (seconds)</i>										
1	35	29	28	77	74	80	109	37	48	43
2	107	72	26	94	50	119	57	12	56	21
3	80	26	41	98	106	23	29	105	46	25
4	38	54	34	56	29	77	53	33	23	29
5	158	45	46	58	102	–	85	87	49	54
6	59	80	47	124	56	65	61	48	47	24
7	96	41	22	59	42	40	21	37	34	20
8	51	42	52	84	53	32	62	83	41	39
9	60	76	33	80	138	84	38	144	24	124
10	72	34	25	94	75	41	57	30	24	49
11	82	45	86	216	92	26	47	53	30	91
12	33	53	48	86	68	62	51	52	27	34
13	43	60	27	99	93	41	45	58	28	35
14	68	60	22	58	42	73	62	87	21	24
15	49	56	31	46	120	45	67	53	52	32
16	28	52	23	47	54	35	43	39	26	18
17	122	40	33	93	41	32	55	32	36	13
18	31	31	30	37	52	30	28	30	20	18
19	30	40	38	60	65	35	32	24	2	24
20	28	32	35	116	30	42	34	35	20	30
21	31	71	20	65	38	36	38	37	32	27
22	54	102	37	52	49	107	58	49	33	45
23	74	135	51	53	46	86	69	67	51	78
24	210	160	59	122	105	67	99	40	30	97
25	31	55	11	62	100	93	53	63	35	67
<i>(b) Efficiency</i>										
1	0.14	0.14	0.18	0.06	0.07	0.06	0.05	0.14	0.10	0.09
2	0.05	0.04	0.19	0.04	0.10	0.03	0.07	0.25	0.09	0.24
3	0.06	0.12	0.12	0.05	0.05	0.17	0.14	0.04	0.09	0.16
4	0.13	0.09	0.15	0.07	0.17	0.06	0.09	0.15	0.17	0.10
5	0.03	0.11	0.11	0.07	0.05		0.06	0.06	0.10	0.07
6	0.07	0.04	0.11	0.03	0.09	0.08	0.07	0.10	0.09	0.21
7	0.05	0.07	0.23	0.07	0.12	0.10	0.24	0.14	0.15	0.25
8	0.10	0.12	0.10	0.05	0.09	0.13	0.08	0.06	0.12	0.13
9	0.08	0.07	0.15	0.05	0.04	0.06	0.11	0.03	0.17	0.03
10	0.07	0.12	0.20	0.04	0.07	0.12	0.07	0.17	0.21	0.10
11	0.06	0.11	0.06	0.02	0.05	0.15	0.09	0.06	0.17	0.04
12	0.12	0.06	0.10	0.06	0.07	0.08	0.10	0.10	0.15	0.09
13	0.09	0.07	0.19	0.05	0.05	0.12	0.11	0.09	0.18	0.11
14	0.07	0.05	0.23	0.07	0.12	0.05	0.06	0.06	0.24	0.17
15	0.10	0.05	0.16	0.09	0.04	0.11	0.06	0.06	0.08	0.09
16	0.14	0.10	0.22	0.09	0.09	0.14	0.12	0.10	0.19	0.22
17	0.03	0.13	0.15	0.04	0.12	0.13	0.09	0.16	0.14	0.38
18	0.16	0.13	0.17	0.11	0.10	0.17	0.14	0.17	0.20	0.28
19	0.13	0.13	0.13	0.08	0.08	0.14	0.16	0.21	2.50	0.21
20	0.18	0.09	0.14	0.03	0.17	0.05	0.12	0.14	0.25	0.13
21	0.16	0.04	0.25	0.06	0.13	0.11	0.11	0.08	0.13	0.11
22	0.09	0.05	0.14	0.08	0.10	0.05	0.09	0.08	0.15	0.09
23	0.07	0.04	0.10	0.08	0.11	0.06	0.07	0.07	0.10	0.06
24	0.02	0.03	0.08	0.03	0.05	0.06	0.04	0.13	0.13	0.05
25	0.13	0.09	0.45	0.08	0.05	0.05	0.09	0.08	0.14	0.07
<i>(c) Effectiveness</i>										
1	1	0.8	1	1	1	1	1	1	1	0.8
2	1	0.6	1	0.8	1	0.8	0.8	0.6	1	1
3	1	0.6	1	1	1	0.8	0.8	0.8	0.8	0.8
4	1	1	1	0.8	1	1	1	1	0.8	0.6
5	0.8	1	1	0.8	1	1	1	1	1	0.8
6	0.8	0.6	1	0.8	1		0.8	1	0.8	1
7	1	0.6	1	0.8	1	0.8	1	1	1	1
8	1	1	1	0.8	1	0.8	1	1	1	1
9	1	1	1	0.8	1	1	0.8	1	0.8	0.8
10	1	0.8	1	0.8	1	1	0.8	1	1	1

Table 8 (continued)

Subject#	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10
11	1	1	1	0.8	1	0.8	0.8	0.6	1	0.8
12	0.8	0.6	1	1	1	1	1	1	0.8	0.6
13	0.8	0.8	1	1	1	1	1	1	1	0.8
14	1	0.6	1	0.8	1	0.8	0.8	1	1	0.8
15	1	0.6	1	0.8	1	1	0.8	0.6	0.8	0.6
16	0.8	1	1	0.8	1	1	1	0.8	1	0.8
17	0.8	1	1	0.8	1	0.8	1	1	1	1
18	1	0.8	1	0.8	1	1	0.8	1	0.8	1
19	0.8	1	1	1	1	1	1	1	1	1
20	1	0.6	1	0.8	1	0.4	0.8	1	1	0.8
21	1	0.6	1	0.8	1	0.8	0.8	0.6	0.8	0.6
22	1	1	1	0.8	1	1	1	0.8	1	0.8
23	1	1	1	0.8	1	1	1	1	1	1
24	0.8	0.8	1	0.8	1	0.8	0.8	1	0.8	1
25	0.8	1	1	1	1	1	1	1	1	1

was too short (2 s) and we decided to ignore this value. The times for these subjects in these exercises were considered as null values. Tables 8b and c show the efficiency and effectiveness of the subjects in the experiment, respectively.

We decided to study the outliers before working with the average data. In order to find the outliers we made box plots (Fig. 7) with the collected data (Table 8a–c). Observing this box plots (Fig. 7a–c) we can observe that there are several outliers (shown in Table 9). The outlier values were eliminated from the collected data. The eliminated values are shown in Table 8a–c in italic font. The descriptive statistics of the final set of data can be found in Table 10a–c. Then, we performed the analysis with these data.

#### 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.

*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 domains 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. For avoiding problems with this issue, we projected the time (hh:mm:ss) in the wall of the room where the experiment took place.
- *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 so, we tried to minimize learning effects.

- *Fatigue effects.* The average time for completing the experiment was smaller than half an hour. With this range of times we believe that fatigue effects hardly exist at all. Furthermore, 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 in 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. Furthermore, the subjects positions during the experiment, did not allow them to communicate.

*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 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 could have been used.
- *Subjects.* Although this experiment was run by experts, we are aware that the number of subjects (25) 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.

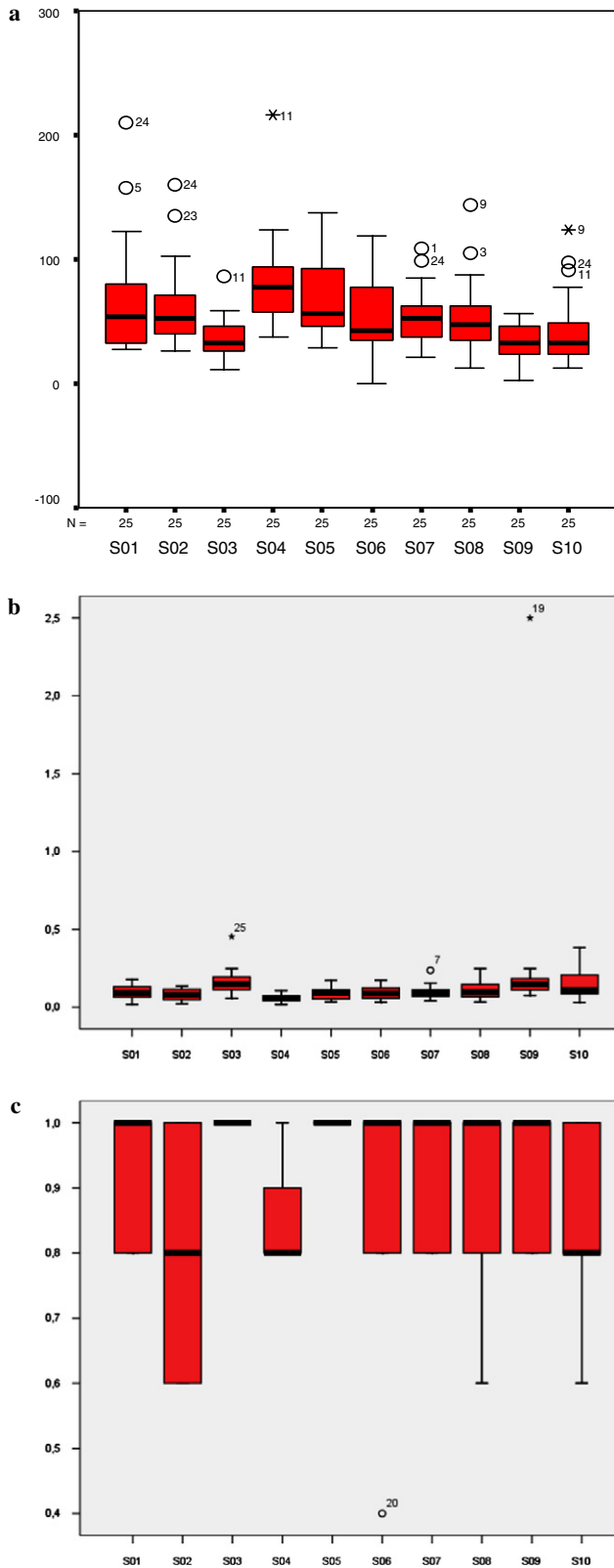


Fig. 7. (a) Box plot of the understanding time; (b) box plot of the efficiency; (c) box plot of the effectiveness.

**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

Table 9

Outliers

Schema	Subject outliers		
	Time	Efficiency	Effectiveness
S01	5.24		
S02	23.24		
S03	11	25	
S04	11		
S05			
S06			20
S07	1.24	7	
S08	3.9		
S09		19	
S10	9.11.24		

the size of the sample data (25 values), which perhaps is not enough for both parametric and non-parametric statistic tests [7]. We will try to obtain bigger sample data through more experimentation.

**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 variable we use is understanding time, i.e., the time each subject spent performing this task, so we consider this variable constructively valid. The construct validity of the measures used for the independent variables is guaranteed by the Distance framework [49] used for their theoretical validation (see Section 5) [53].

Although, we know that several aspects threaten the validity of the results, we have tried to alleviate them by different means. In this section we have discuss the problems that could affect the results of the experiment and how we tried to solve them. We know that even though we put a lot of effort in alleviate the threats, some of them can affect the results and could lessen the strength of the results. As we have made a family of experiments and we have obtained the same results in all the experiments, we think that those threats have had a small impact on the results. We plan to make more experiments and case studies varying some empirical settings to get more conclusive results.

**5.3.2.3. Analysis and interpretation.** We used the data collected in order to test the hypotheses previously formulated. 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 11a shows the results obtained for the correlation between each of the metrics and the time used by each subject (on each schema) in performing the tasks. Tables 11b and c show the results of the correlation analyses between metrics and efficiency and effectiveness, respectively.



Table 10  
Descriptive statistics

	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10
<i>(a) Understanding time</i>										
Average	56.61	52.00	34.13	75.83	68.80	57.13	49.78	47.22	34.71	34.05
Minimum	28	26	11	37	29	23	21	12	20	13
Maximum	122	102	59	124	138	119	85	87	56	78
Deviation	27.14	18.78	11.82	25.05	30.15	27.39	15.38	19.87	11.31	16.43
<i>(b) Efficiency</i>										
Average	0.02	0.03	0.06	0.02	0.04	0.03	0.04	0.03	0.08	0.03
Minimum	0.18	0.14	0.25	0.11	0.17	0.17	0.16	0.25	0.25	0.38
Maximum	0.05	0.04	0.05	0.02	0.04	0.04	0.03	0.05	0.05	0.09
Deviation	0.02	0.03	0.06	0.02	0.04	0.03	0.04	0.03	0.08	0.03
<i>(c) Effectiveness</i>										
Average	0.8	0.6	1	0.8	1	0.8	0.8	0.6	0.8	0.6
Minimum	1	1	1	1	1	1	1	1	1	1
Maximum	0.1	0.18	0	0.09	0	0.1	0.1	0.15	0.1	0.15
Deviation	0.8	0.6	1	0.8	1	0.8	0.8	0.6	0.8	0.6

Table 11  
Results of the experiment

Metric	NDC	NBC	NC	RBC	NAFC	NADC	NABC	NA	NH	DHP	RSA
<i>(a) Understanding time</i>											
Correlation	0.619	0.877	0.835	0.772	0.544	-0.215	0.756	0.745	0.773	0.687	-0.018
<i>p</i> -value	0.056	0.001	0.003	0.009	0.104	0.551	0.011	0.013	0.009	0.028	0.960
<i>(b) Efficiency</i>											
Correlation	-0.514	-0.853	-0.823	-0.723	-0.169	-0.006	-0.470	-0.515	-0.896	-0.482	0.164
<i>p</i> -value	0.129	0.002	0.003	0.018	0.641	0.987	0.171	0.128	0.000	0.159	0.651
<i>(c) Effectiveness</i>											
Correlation	-0.246	-0.062	-0.095	-0.012	0.302	-0.086	0.144	0.067	-0.346	0.090	0.220
<i>p</i> -value	0.493	0.865	0.794	0.973	0.396	0.812	0.691	0.854	0.328	0.804	0.542

Analysing Table 11a, we can conclude that there exists a correlation between the understanding time used (understandability of the schemas) and the metrics NBC, NC, RBC, NABC, NA, NH and DHP (the *p*-value is lower than or equal to  $\alpha = 0.05$ ) and that the metrics NDC, NAFC, NADC and RSA are not correlated with time.

Analysing Table 11b, we can see that the metrics NBC, NC, RBC and NH are correlated with the efficiency of the experimental subjects when solving the experimental tasks. This correlation is an inverse relationship, that is to say that the lower the value of these metric is, the higher is the efficiency of the subjects.

On the other hand we can see in Table 11c that none of the proposed metrics are correlated with the effectiveness of the subjects when dealing with conceptual data warehouse schemas.

### 5.3.3. Conclusions of the complete experimental work

Table 12 summarizes all the empirical work we have performed with the metrics for conceptual data warehouse models. After all the experimental work, we can conclude that the metrics NBC, NC, RBC, NABC, NA, NH and DHP (Number of Base Classes, Number of Classes, Ratio of Base Classes, Number of Attributes of Base Classes, Number of Attributes, Number of Hierarchy relationships and maximum Depth of the Hierarchy Path) seem to be correlated with the understandability of data warehouse conceptual models.

Regarding efficiency and effectiveness, in the last experiment we have found that there is a high inverse relationship between NBC, NC, RBC and NH metrics with the efficiency of the subjects working with data warehouse conceptual schemas. It seems that the understandability and

Table 12  
Results summary of all the experimental work (✓ means that there is a relationship between understandability and the metric. ✓✓ means that there is also a relationship between efficiency and the metric)

	NDC	NBC	NC	RBC	NAFC	NADC	NABC	NA	NH	DHP	RSA
1st exp		✓	✓	✓			✓	✓	✓	✓	
2nd exp		✓	✓	✓			✓	✓	✓	✓	
3rd exp		✓✓	✓✓	✓✓			✓	✓	✓✓	✓	

the efficiency are related to the number of classes of the DW conceptual schema and the number of the hierarchy paths defined in dimensions.

Although we have found encouraging results, we must go on with the experimental process in order to confirm the influence of the proposed metrics in the efficiency and effectiveness in using data warehouses. After these new experiments, we will move into the next step of our method (See Fig. 3) which means that the proposed metrics must be applied in real world projects prior to the final acceptance.

With these metrics, designers could choose from alternative semantically equivalent schemata, basing on objective indicators. They could measure the schemata obtained with different design techniques or different design decisions and choose which one of them could fit their goals. Also, they could use the metrics to predict the effort of understanding that the users, who deal with a schema, have to burden.

## 6. Conclusions and future research

As many strategic decisions taken in companies are based on the data stored in data warehouses (DW), assuring the quality of these DWs is absolutely crucial for companies. One aspect to assure their quality is to guarantee the quality of the models used in their design (conceptual, logical and physical).

In this paper, we have proposed a set of metrics in order to assure the quality of the conceptual schema used in the early stages of a DW design. These metrics will help us measure the understandability and the efficiency of designers and users in working with the schemas. The proposed metrics have been theoretically validated (by using both the Briand and Poels frameworks) to guarantee that they have been correctly defined and that they will help us measure what they are intended for. Then, we have also presented a set of experiments we have accomplished in order to proof the validity of the proposed metrics. After these experiments we can conclude that several metrics are correlated with the understandability of the models (mainly those measuring the number of elements in the conceptual schema such as the number of classes, associations, attributes, and so on) and with the efficiency of the subjects when dealing with those models (those measuring the number of classes, dimensions, and the number of hierarchy levels defined in dimensions).

Our immediate future work is to apply the valid metrics in real world projects in order to pass into the next step of our method, which will allow us the final acceptance of the proposed metrics. We are also currently working in defining new metrics to measure the effectiveness of DW conceptual schemas. On having accepted a set of metrics to measure all these aspects, we plan to define a set of quality indicators which helps and guide the designer in designing DWs. Another relevant issue would be to be able to define

the correct metrics thresholds under which several design options can be taken. Another further aspect we will deal with is the traceability of metrics as one conceptual schema can be transformed into several logical schemas (pure star schema, snowflake, normalizing only come dimensions, and so on).

## Acknowledgements

This research is part of the CALIPO project (TIC2003-07804-C05-03) and the METASIGN project (TIN2004-00779), supported by Dirección General de Investigación of the Ministerio de Ciencia y Tecnología. This research is also part of the DADASMECA project (GV05/220) supported by the Valencia Ministry of Enterprise, and the CALIA Project supported by the University of Castilla – Mancha. We would like to thank the anonymous reviewers for their invaluable feedback. Also, we would like to thank all the people at University of Alicante who kindly volunteered to take part in the experiment, specially accomplished for this paper.

## References

- [1] A. Abelló, J. Samos, F. Saltor, A framework for the classification and description of multidimensional data models, in: 12th International Conference on Database and Expert Systems Applications (DEXA' 01), Springer-Verlag, Munich (Germany), 2001.
- [2] A. Abelló, J. Samos, F. Saltor, YAM2 (Yet Another Multidimensional Model): An Extension of UML, in: International Database Engineering and Applications Symposium (IDEAS 2002), IEEE Computer Society, Edmonton (Canada), 2002, pp. 172–181.
- [3] A.J. Albrecht, S.H. Gaffney, Source function, source lines of code and development effort prediction: a software science validation, *IEEE Transactions on Software Engineering* 9 (1983) 639–648.
- [4] V. Basili, F. Shull, F. Lanubile, Building knowledge through families of experiments, *IEEE Transactions on Software Engineering* 25 (4) (1999) 435–437.
- [5] V. Basili, D. Weiss, A Methodology for Collecting Valid Software Engineering Data, *IEEE Transactions on Software Engineering* 10 (1984) 728–738.
- [6] M. Blaschka, C. Sappia, G. Höfling, B. Dinter, Finding your way through multidimensional data models, 9th International Conference on Database and Expert Systems Applications (DEXA '98), Springer-Verlag, Vienna (Austria), 1998, pp. 198–203.
- [7] L. Briand, K. El Emam, S. Morasca, Theoretical and empirical validation of software product measures, Technical Report ISERN-95-03, International Software Engineering Research Network, 1995.
- [8] L. Briand, S. Morasca, V. Basili, Property-Based Software Engineering Measurement, *IEEE Transactions on Software Engineering* 22 (1) (1996) 68–86.
- [9] L. Briand, J. Wüst, H. Lounis, A Comprehensive Investigation of Quality Factors in Object-Oriented Designs: an Industrial Case Study, International Software Engineering Research Network (1998).
- [10] F. Brito e Abreu, R. Carapuça, Object-Oriented Software Engineering: measuring and controlling the development process, in: 4th International Conference on Software Quality, McLean (USA), 1994.
- [11] C. Calero, Definition of a set of metrics for relational, object-oriented and active databases maintainability, *Computer Science, University of Castilla-La Mancha, Ciudad Real (Spain)*, 2001.

- [12] C. Calero, M. Piattini, M. Genero, Method for obtaining correct metrics, 3rd International Conference on Enterprise and Information Systems (ICEIS'2001), 2001, pp. 779–784.
- [13] G. Cantone, P. Donzelli, Production and maintenance of software measurement models, *Journal of Software Engineering and Knowledge Engineering* 5 (2000) 605–626.
- [14] S. Chidamber, C. Kemerer, A Metrics Suite for Object Oriented Design, *IEEE Transactions on Software Engineering* 20 (6) (1994) 476–493.
- [15] C. Eick, A methodology for the design and transformation of conceptual schemas, in: 17th International Conference on Very Large Data Bases, Barcelona (Spain), 1991, pp. 25–34.
- [16] L. English, *Information Quality Improvement: Principles, Methods and Management*, Brentwood, Information Impact International, Inc., 1996.
- [17] N. Fenton, S. Pfleeger, *Software Metrics: A Rigorous Approach*, Chapman & Hall, London, 1997.
- [18] R.L. Flood, E.R. Carson, *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science*, Springer, 1993.
- [19] M. Genero, *Defining and Validating Metrics for Conceptual Models*, Department of Computer Science, University of Castilla-La Mancha, Ciudad Real (Spain), 2002.
- [20] M. Genero, J. Olivas, M. Piattini, F. Romero, Using metrics to predict OO information systems maintainability, in: 13th International Conference Advanced Information Systems Engineering (CAiSE'01) (2001), pp. 388–401.
- [21] M. Golfarelli, D. Maio, S. Rizzi, The Dimensional Fact Model: A Conceptual Model for Data Warehouses, *International Journal of Cooperative Information Systems (IJCIS)* 7 (1998) 215–247.
- [22] M. Golfarelli, S. Rizzi, A methodological framework for data warehouse design, in: 1st International Workshop on Data Warehousing and OLAP (DOLAP '98), Maryland (USA), 1998, pp. 3–9.
- [23] R. Gray, B. Carey, N. McGlynn, A. Pengelly, Design metrics for database systems, *BT Technology* 9 (1991).
- [24] M. Halstead, *Elements of Software Science*, Elsevier-North Holland, New York, 1977.
- [25] B. Husemann, J. Lechtenbörger, G. Vossen, Conceptual data warehouse design, in: 2nd, International Workshop on Design and Management of Data Warehouses (DMDW 2000), Stockholm (Sweden), 2000, pp. 3–9.
- [26] IEEE, IEEE Std 1061-1998 IEEE Standard for a Software Quality Metrics Methodology, 1998.
- [27] W.H. Inmon, *Building the Data Warehouse*, John Wiley and Sons, USA, 2003.
- [28] ISO/IEC, ISO/IEC 15504 TR2:1998, Software Process Assessment – Part 2: A Reference Model for Processes and Process Capability, in: I.I. JTC1/SC7, (Ed.), *International Organization for Standardization*, 1998.
- [29] ISO/IEC, 9126-1: Software Engineering – Product quality – Part 1: Quality model., 2001.
- [30] ISO/IEC, ISO 15939: Software Engineering – Software Measurement Process, 2002.
- [31] ISO/IEC, ISO/IEC 90003, Software and Systems Engineering – Guidelines for the Application of ISO/IEC 9001:2000 to Computer Software, International Standards Organization, Geneva, Switzerland, 2004.
- [32] M. Jarke, M. Lenzerini, Y. Vassiliou, P. Vassiliadis, *Fundamentals of Data Warehouses*, Springer-Verlag, 2002.
- [33] M. Jeusfeld, C. Quix, and M. Jarke, Design and Analysis of Quality Information for Data Warehouses, in: 17th International Conference on Conceptual Modeling (ER'98), Singapore, 1998.
- [34] N. Juristo, A. Moreno, *Basics of Software Engineering Experimentation*, Kluwer Academic Publishers, 2001.
- [35] S. Kesh, Evaluating the Quality of Entity Relationship Models, *Information and Software Technology* 37 (1995) 681–689.
- [36] R. Kimbal, M. Ross, *The Data Warehouse Toolkit*, John Wiley and Sons, 2002.
- [37] B. Kitchenham, S. Pfleeger, L. Pickard, P. Jones, D. Hoaglin, K. El Emam, J. Rosenberg, Preliminary Guidelines for Empirical Research in Software Engineering, *IEEE Transactions on Software Engineering* 28 (2002) 721–734.
- [38] G.J. Klir, D. Elias, *Architecture of Systems Problem Solving*, Prentice Hall, New York, 2003.
- [39] J. Lechtenbörger, G. Vossen, Multidimensional Normal Forms for Data Warehouse Design, *Information Systems* 28 (2003) 415–434.
- [40] W. Lehner, J. Albrecht, H. Wedekind, Normal forms for multidimensional databases, in: 10th International Conference on Scientific and Statistical Database Management (SSDBM), IEEE Press, 1998, pp. 63–72.
- [41] M. Lorenz, J. Kidd, *Object-Oriented Software Metrics: A Practical Guide*, Prentice Hall, Englewood Cliffs (New Jersey), 1994.
- [42] S. Luján-Mora, J. Trujillo, I.-Y. Song, Extending UML for multidimensional modeling, in: 5th International Conference on the Unified Modeling Language (UML 2002), LNCS 2460, Dresden (Germany), 2002, pp. 290–304.
- [43] M. Marchesi, OOA metrics for the unified modeling language, in: 2nd Euromicro Conference on Software Maintenance and Reengineering, 1998, pp. 67–73.
- [44] T. McCabe, A Software Complexity Measure, *IEEE Transaction on Software Engineering* 2 (1976) 308–320.
- [45] J. McGarry, D. Card, C. Jones, B. Layman, E. Clark, J. Dean, F. Hall, *Practical Software Measurement Objective Information for Decision Makers*, Wiley, 2002.
- [46] D. Moody, Metrics for evaluating the quality of entity relationship models, 17th International Conference on Conceptual Modelling (ER '98), Singapore, 1998, pp. 213–225.
- [47] OMG, *OMG Unified Modeling Language Specification; versión 2.0*, Object Management Group, 2005.
- [48] S. Pfleeger, B. Kitchenham, *Principles of Survey Research. Part 1: Turning Lemons into Lemonade*. ACM Sigsoft, *Software Engineering Notes* 26 (6) (2001) 16–18.
- [49] G. Poels, G. Dedene, DISTANCE: A Framework for Software Measure Construction, Research Report DTEW9937, Dept. Applied Economics Katholieke Universiteit Leuven, Belgium, 1999, p. 46.
- [50] C. Sapia, On modeling and predicting query behaviour in olap systems, International Workshop on Design and Management of Data Warehouses (DMDW '99), Heidelberg (Germany), 1999, pp. 1–10.
- [51] C. Sapia, M. Blaschka, G. Höfling, B. Dinter, Extending the E/R model for the multidimensional paradigm, 1st International Workshop on Data Warehouse and Data Mining (DWDW' 98), Springer-Verlag, Singapore, 1998, pp. 105–116.
- [52] SEI, *Capability Maturity Model Integration (CMMI)*, version 1.1, 2002.
- [53] M. Serrano, Definition of a Set of Metrics for Assuring Data Warehouse Quality, University of Castilla, La Mancha (Spain), 2004.
- [54] M. Serrano, C. Calero, M. Piattini, Validating metrics for data warehouses, *IEEE Proceedings SOFTWARE* 149 (2002) 161–166.
- [55] M. Serrano, C. Calero, J. Trujillo, S. Lujan, M. Piattini, Empirical validation of metrics for conceptual models of data warehouse, 16th International Conference on Advanced Information Systems Engineering (CAISE'04), Riga, Latvia, 2004, pp. 506–520.
- [56] M. Serrano, C. Calero, J. Trujillo, S. Lujan, M. Piattini, Empirical validation of metrics for data warehouses, 4th ASERC Workshop on Quantitative and Soft Computing Based Software Engineering (QSSE 2004), Banff, Alberta (Canada), 2004.
- [57] S. Si-Said, N. Prat, Multidimensional Schemas Quality: Assessing and Balancing Analyzability and Simplicity. in: M.A.a.P. Jeusfeld, O., (Ed.), *ER 2003 Workshops*, 2003, pp. 140–151.
- [58] P. Suppes, M. Krantz, R. Luce, A. Tversky, *Foundations of Measurement*, Academic Press, New York, 1989.
- [59] J. Trujillo, M. Palomar, J. Gómez, I.-Y. Song, Designing Data Warehouses with OO Conceptual Models. *IEEE Computer, Special issue on Data Warehouses* 34 (2001) 66–75.

- [60] N. Tryfona, F. Busborg, J. Christiansen, starER: A Conceptual Model for Data Warehouse Design, ACM 2nd International Workshop on Data Warehousing and OLAP (DOLAP' 99), ACM, Missouri (USA), 1999, pp. 3–8.
- [61] R. Van Solingen, E. Berghout, The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development, McGraw-Hill, 1999.
- [62] P. Vassiliadis, Data Warehouse Modeling and Quality Issues, National Technical University of Athens, Athens(Greece), 2000.
- [63] E. Weyuker, Evaluating Software Complexity Measures, IEEE Transactions on Software Engineering 14 (9) (1988) 1357–1365.
- [64] S. Whitmire, Object Oriented Design Measurement, John Wiley & Sons, Inc., 1997.
- [65] C. Wohlin, P. Runeson, M. Höst, M. Ohlson, B. Regnell, A. Wesslén, Experimentation in Software Engineering: An Introduction, Kluwer Academic Publishers, 2000.
- [66] H. Zuse, A Framework of Software Measurement, Walter de Gruyter, Berlin, 1998.