# Towards Data Warehouse Quality Metrics

Coral Calero
ALARCOS Research Group
University of Castilla-La Mancha (Spain)
Rda. Calatrava s/n 13071 Ciudad Real - Spain
ccalero@inf-cr.uclm.es

Mario Piattini
ALARCOS Research Group
University of Castilla-La Mancha (Spain)
Rda. Calatrava s/n 13071 Ciudad Real - Spain
mpiattin@inf-cr.uclm.es

Carolina Pascual
ALARCOS Research Group
University of Castilla-La Mancha (Spain)
Rda. Calatrava s/n 13071 Ciudad Real - Spain
ccoimbra@proyectos.inf-cr.uclm.es

Manuel A. Serrano
ALARCOS Research Group
University of Castilla-La Mancha (Spain)
Rda. Calatrava s/n 13071 Ciudad Real - Spain
mserrano@inf-cr.uclm.es

## Abstract

Organizations are adopting datawarehouses to manage information efficiently as “the” main organizational asset. It is essential that we can assure the information quality of the data warehouse, as it became the main tool for strategic decisions. Information quality depends on presentation quality and the data warehouse quality. This last includes the multidimensional model quality. In the last years different authors have proposed some useful guidelines to design multidimensional models, however more objective indicators are needed to help designers and managers to develop quality datawarehouses. In this paper a first proposal of metrics for multidimensional model quality is shown together with their formal validation.

## 1 Introduction

Nowadays organizations can store vast amounts of data obtained at a relatively low cost, however these data fail to provide information [GAR98]. To solve this problem, organizations are adopting a data warehouse, which is defined as a “collection of subject-oriented, integrated, non-volatile data that supports the management decision process” [INM96]. Data warehouses have become the key trend in corporate computing in the last years, since they provide managers with the most accurate and relevant information to improve strategic decisions. Jarke et al. [JAR00] forecast a 12 Millions American dollars for the data warehouse market.

Different life cycles and techniques have been proposed for data warehouse development [HAM96] [KEL97] [KIM98]. However the development of a data warehouse is a difficult and very risky task. It is essential that we can assure the information quality of the data warehouse as it became the main tool for strategic decisions [ENG99].

Information quality of a data warehouse comprise data warehouse system quality and data presentation quality (see figure 1). In fact, it is very important that data in the data warehouse reflects correctly the real world, but it is also very important that data can be easily understood. In data warehouse system quality, as in an operational database [PIA00], three different aspects could be considered: DBMSs quality, data model quality (both conceptual and logical) and data quality.

In order to assess DBMS quality we can use an international standard like [ISO98], or some of the existing product comparative studies. This type of quality should be addressed in the product selection stage of the data warehouse life cycle.

Data quality is composed by the data definition quality (degree to which data definition accurately describes the meaning of the real-world entity type or fact-type that the data represents. Also meets the needs of all information customers to understand the data they use), the data content quality (degree to which data values accurately represent the characteristics of the real-world entity or fact and meet the need of the information costumers to perform their jobs effectively) and the data presentation quality (try to capture the degree in which the format presented is intuitive for the use to be made of the information)

**Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2001)**
Interlaken, Switzerland, June 4, 2001
(D. Theodoratos, J. Hammer, M. Jeusfeld, M. Staudt, eds.)

http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-39/

[ENG98]. This kind of quality must address mostly in the extraction, filtering, cleaning and cleansing, synchronization, aggregation, loading, etc. activities of the life cycle. In the last years very interesting techniques have been proposed to assure data quality [BOU00].

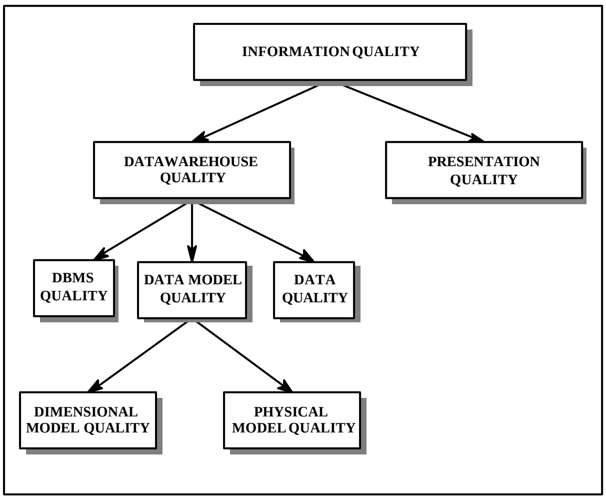

Figure 1 Information and data warehouse quality

Last, but not least, data warehouse model quality has a great influence in the overall information quality. The designer has to choose the physical tables, processes, indexes and data partitions, representing the logical data warehouse and facilitating its functionality [JAR00]. Two different aspects should be considered: dimensional data model quality and physical data model quality. In fact these two are often considered as different stages in the data warehouse life cycle.

Dimensional data model is usually designed using the star schema modeling facility, which allows good response times and an easy understanding of data and metadata for both users and developers [KIM98].

Different techniques have also been researched for optimizing physical data models [HAR96] [LAB97].

Our work focuses on dimensional data model quality. Different authors have suggested interesting recommendations for achieving a "good" dimensional data model [KIM98] [ADM98] [INM96]. However quality criteria are not enough on their own to ensure quality in practice, because different people will generally have different interpretations of the same concept. According to the Total Quality Management (TQM) literature, measurable criteria for assessing quality are necessary to avoid "arguments of style" [BOM97]. The objective should be to replace intuitive notions of design "quality" with formal, quantitative measures in order to reduce subjectivity and bias in the evaluation process. However, for data modeling to progress from a "craft" to an engineering discipline, the desirable qualities of data models need to be made explicit [LIN94]. A metric is away of measuring a quality factor in a consistent and objective manner

Metrics could be used to build prediction systems for database projects, to understand and improve software development and maintenance projects, to maintain the quality of the systems, highlighting problematic areas, and to determine the best ways to help practitioners and researchers in their work.

The final goal of our work is to define a set of metrics for assuring data warehouse quality by means of measuring the dimensional data model quality. In the next section we will present the method we use for defining correct metrics. A first proposal of metrics for data warehouses will be described in section 3 and an example of the proposed metrics will be shown in section 4. Section 5 will present the formal validation of the metrics and conclusions and future work will come in the last section.

## 2 Defining Valid Metrics

Metrics definition must be done in a methodological way and it is necessary to follow a number of steps to ensure the reliability of the proposed metrics. Figure 2 presents the method we follow for the metrics proposal [CAL01].

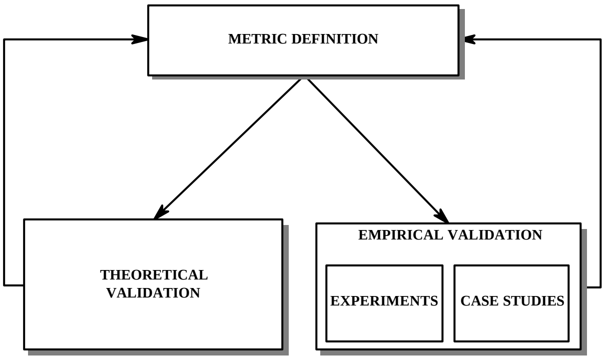

Figure 2. Steps followed in the definition and validation of metrics

In this figure we have three main activities:

- **Metrics definition.** The first step is the proposal of metrics. This definition is made taking into account the specific characteristics of the system we want to measure and the experience of designers of these systems. A goal-oriented approach, as GQM (Goal-Question-Metric) [BAS84] can also be very useful in this step.
- **Theoretical validation.** The second step is the formal validation of the metrics. The formal validation helps us to know when and how apply the metrics. There are two main tendencies in metrics validation: the frameworks based on axiomatic approaches [WEY88] [BRI96] [MOR97] and the ones based on the measurement theory [WIT97] [ZUS98]. The strength of measurement theory is the formulation of empirical conditions from which we can derive hypothesis of reality. The final information when applying this kind of

frameworks in to know to which scale a metric pertains and based on this information we can know which statistics and which transformations can be done with the metric.
- **Empirical validation.** The goal of this step is to prove the practical utility of the proposed metrics. Although there are various ways of performing this step, basically we can divide the empirical validation into experimentation and case studies [BAS99] [FEN97].

As shown in figure 2, the process of defining and validating metrics is evolutionary and iterative. As a result of the feedback, metrics could be redefined based on discarded theoretical or empirical validations. At the moment, we have finished only a first iteration of metrics definition and theoretical validation for data warehouse quality.

This work only considers the two steps related with definition and formal validation. It is clear that these are only a first approach because it is fundamental to made empirical validation in order to prove which of all of the proposed metrics are useful in the real world rejecting the ones that do not prove its uselfulness.

## 3 Metrics for Data Warehouses

Taking into account the characteristics exposed previously, we will propose the following metrics for data warehouses. As some metrics can be applied at table, star and schema level, we present them separately.

### 3.1 Table Metrics

In the last years we have researched different metrics for assuring relational database quality [PIA01]. Two of these metrics could be useful for data warehouses:

- **NA(T).** Number of attributes of a table.
- **NFK(T).** Number of foreign keys of a table.

### 3.2 Star Metrics

- **NDT(S).** Number of dimension tables of a star.
- **NT(S).** Number of tables of a star, which corresponds to the number of dimension tables added the fact table.

  $NT(S) = NDT(S) + 1$
- **NADT(S).** Number of attributes of dimension tables of a star.
- **NAFT(S).** Number of attributes plus the number of foreign keys of a fact table.

  $NAFT(S) = NA(FT) + NFK(FT)$

  Where FT is the fact table of the star S.
- **NA(S).** Number of attributes of a star.

  $NA(S) = NAFT(FT) + NADT(S)$

  Where FT is the fact table of the star S.
- **NFK(S).** Number of foreign keys of a star.

$$NFK(S) = NFK(FT) + \sum_{i=1}^{NDT} NFK(DT_i)$$

Where FT is the fact table of the star S
and Dti is the dimensional table number i of the star S

- **RSA(S).** Ratio of star attributes. Quantity of the number of attributes of dimension tables per number of attributes of the fact table of the star.

$$RSA(S) = \frac{NADT(S)}{NAFT(FT)}$$

Where FT is the fact table of the star S.

- **RFK(S).** Ratio of foreign keys. Quantity of the fact table attributes which are foreign key.

$$RFK(S) = \frac{NFK(FT)}{NAFT(FT)}$$

Where FT is the fact table of the star S

### 3.3 Schema Metrics

- **NFT(Sc).** Defined as a number of fact tables of the schema.
- **NDT(Sc).** Number of dimension tables of the schema.
- **NSDT(Sc).** Number of shared dimension tables. Number of dimension tables shared for more than one star of the schema.
- **NT(Sc).** Number of tables. Number of the fact tables plus the number of dimension tables of the schema.
- **NAFT(Sc).** Number of attributes of fact tables of the schema.

$$NAFT(Sc) = \sum_{i=1}^{NFT} NA(FT_i)$$

Where FTi is 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 DTi is the dimensional table i of the schema Sc

- **NASDT(Sc).** Number of attributes of shared dimension tables of the schema.

$$NASDT(Sc) = \sum_{i=1}^{NSDT} NA(DTi)$$

Where DTi is 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 FTi is the fact table i of the schema Sc

- **RSDT(Sc).** Ratio of Shared Dimension Tables. Quantity of dimension tables, which belong to more than one star.

$$RSDT(Sc) = \frac{NSDT(Sc)}{NDT(Sc)}$$

- **RT(Sc).** Ratio of tables. Quantity of dimension tables per fact table.

$$RT(Sc) = \frac{NDT(Sc)}{NFT(Sc)}$$

- **RScA(Sc).** Ratio of Schema Attributes. Number of attributes in dimension tables per attributes in fact tables.

$$RScA(Sc) = \frac{NADT(Sc)}{NAFT(Sc)}$$

- **RFK(Sc).** Ratio of foreign keys. Quantity of attributes that are foreign key.

$$RFK(Sc) = \frac{NFK(Sc)}{NA(Sc)}$$

- **RSDTA(Sc).** Ratio of Shared Dimension Tables Attributes. Number of attributes of the schema that are shared.

$$RSDTA(Sc) = \frac{NASDT(Sc)}{NA(Sc)}$$

As we can see we have proposed a large set of metrics, now we must validate, formal and empirically, and pick up the metrics that can be useful in order to measure the quality of a data warehouse schema.

## 4 Example

Figure 3 shows an example of a Data Warehouse [ADA98]. The values for the metrics are shown in tables 1, 2 and 3.

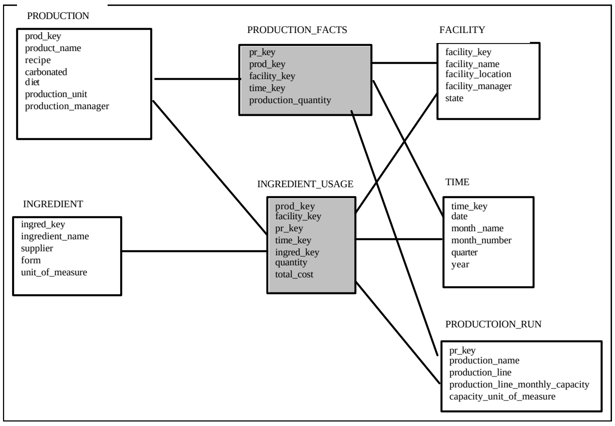

Figure 3. Example of a data warehouse star design [ADA98]

| | NA | NFK |
|---|---|---|
| PRODUCT | 7 | 0 |
| INGREDIENT | 5 | 0 |
| FACILITY | 5 | 0 |
| TIME | 6 | 0 |
| PRODUCTION-RUN | 5 | 0 |
| PRODUCTION-FACTS | 5 | 4 |
| INGREDIENT-USAGE | 7 | 5 |

Table 1. Values for table metrics

| | Production-Facts | Ingredient-Usage |
|---|---|---|
| NA | 28 | 35 |
| NFK | 4 | 5 |
| RSA | 23/5 | 28/7 |
| NDT | 4 | 5 |
| NT | 5 | 6 |
| NADT | 23 | 28 |
| NAFT | 5 | 7 |
| RFK | 4/28 | 5/35 |

Table 2. Values for star metrics

| Metric | Value |
|---|---|
| NA | 40 |
| NFK | 9 |
| NDT | 5 |
| NT | 7 |
| NADT | 28 |
| NAFT | 12 |
| RFK | 9/40 |
| NFT | 2 |
| NSDT | 4 |
| NASDT | 23 |
| RSDT | 4/5 |
| RT | 5/2 |
| RScA | 28/12 |
| RSDTA | 23/40 |

Table 3. Values for datawarehouse schema metrics

## 5 Metrics formal validation

In this section we will present the metrics formal validation made using the formal framework proposed by [ZUS98]. 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. 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 5.

The formal framework of [ZUS98] works with three main mathematical structures, depending of which one of this structures a metric accomplishes, we will be able to characterize it in a scale. These three structures (table 4) are: the extensive structure, the independence conditions and the modified relation of belief. All the details about these three structures and the complete formal framework can be found in [ZUS98].

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 (that describes the properties of the software measure clearly) 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 does not accomplish the extensive structure, neither the independence conditions but it accomplishes the modified relation of belief, can be characterized "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).

### 5.1 NFK Metric Formal Validation

The NFK measure is a mapping: NFK: T -> ℜ such that the following holds 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 characterise the combination rule for NFK as:

NFK($T_i$o$T_j$) = NFK($T_i$) + NFK($T_j$) -v

| MODIFIED EXTENSIVE STRUCTURE |
|---|
| **Axiom1**: (A, • >=) (weak order)<br>**Axiom2**: A1 o A2 • >= A1 (positivity)<br>**Axiom3:** A1 o (A2 o A3) ≈ (A1 o A2) o A3 (weak associativity)<br>**Axiom4**: A1 o A2 ≈ A2 o A1 (weak commutativity)<br>**Axiom5**: A1 • >= A2 ⇒ A1 o A • >= A2 o A (weak monotonicity)<br>**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:<br>A1 • >= A2, and A2 • >= A3 ⇒ A1 • >= A3<br>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<br>**C2**: A1 ≈ A2 ⇔ A1 o A ≈ A2 o A and A1 ≈ A2 ⇔A o A1 ≈ A o A2<br>**C3**: A1 • >= A2 ⇒ A1 o A • >= A2 o A, and A1 • >= A2 ⇒ A o A1 • >= A o A2<br>**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<br>A1 • >= A2 and not (A2 • >= A1). |
| **MODIFIED RELATION OF BELIEF** |
| **MRB1**: ∀ A, B ∈ ℑ: A • >= B or B • >= A (completeness)<br>**MRB2**: ∀ A, B, C ∈ ℑ: A • >= B and B • >= C ⇒ A • >= C (transitivity)<br>**MRB3**: ∀ A ⊇ B ⇒ A • >= B (dominance axiom)<br>**MRB4**: ∀ (A ⊃ B, A ∩ C = φ) ⇒ (A • >= B ⇒ A U C • > B U C) (partial monotonicity)<br>**MRB5**: ∀ A ∈ ℑ: A • >= 0 (positivity) |

Table 4. Summary of the mathematical structures of the Zuse's formal framework

**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 fulfils the first axiom.

The positivity axiom (axiom 2) is not verified by the metrics own definition (when v is distinct of zero). For example, in figure 3 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 4, 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.

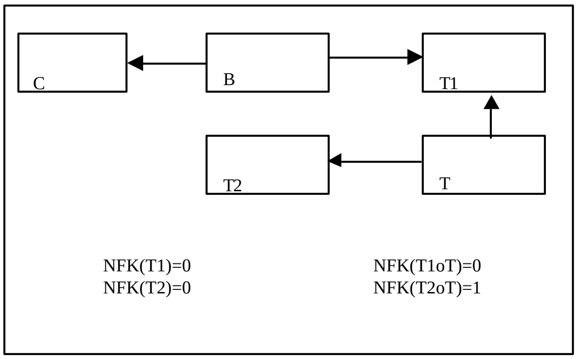

Figure 4. Some NFK values

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 5, we can assure that NFK does not accomplish the Archimedean axiom (NFK(R3)>NFK(R4) and NFK(R1oR3)<NFK(R2oR4)).

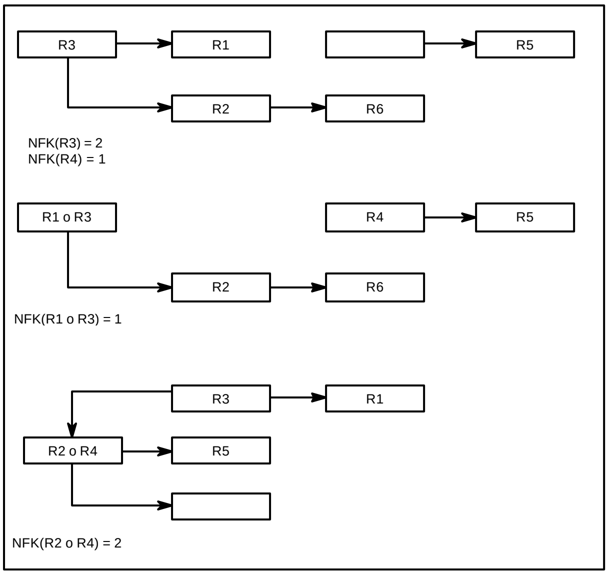

Figure 5. NFK does not accomplish the Archimedean axiom.

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

## NFK and the independence conditions.

The metric does not accomplish the first condition, seeing figure 5, R2 and R4 have a value equal to 1 (NFK(R2)=1, NFK(R4)=1). If we combine this 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 fulfil the fifth axiom of the extensive structure and if it does not accomplish the third it cannot accomplish the fourth one. 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 is also accomplished because if A⊃B then NFK(A)>NFK(B), if there are not common foreign keys between A and C, it cannot be neither between B and C, and then the fourth conditions will be accomplished because NFK(AoC)>NFK(BoC). The last condition, positivity, is also 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.

As we said previously, the result of the formal validation of the rest of metrics in the Zuse formal framework are summarized in table 5. It is necessary to point out that following [ZUS98] all the metrics defined as a percentage can be characterized in the absolute scale.

| | NA | NFK | NDT | NT | NAD T | NAFT | NFT | NSDT | NASDT |
|---|---|---|---|---|---|---|---|---|---|
| **Ax 1** | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| **Ax 2** | N | Y | Y | Y | Y | N | Y | N | Y |
| **Ax 3** | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| **Ax 4** | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| **Ax 5** | N | N | N | Y | N | N | Y | N | Y |
| **Ax 6** | N | N | N | Y | N | N | Y | N | Y |
| **Ind C 1** | N | N | N | | N | N | | N | |
| **Ind.C 2** | N | N | N | | N | N | | N | |
| **Ind C 3** | N | N | N | | N | N | | N | |
| **Ind C 4** | N | N | N | | N | N | | N | |
| **MRB1** | Y | Y | Y | | Y | Y | | Y | |
| **MRB2** | Y | Y | Y | | Y | Y | | Y | |
| **MRB3** | Y | Y | Y | | Y | Y | | Y | |
| **MRB4** | Y | Y | Y | | Y | Y | | Y | |
| **MRB5** | Y | Y | Y | | Y | Y | | Y | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **SCALE** | **AB<br>ORD** | **AB<br>ORD** | **AB<br>ORD** | **RAT** | **AB<br>ORD** | **AB<br>ORD** | **RAT** | **AB<br>ORD** | **RAT** |
| **RSA, RFK, RSDT, RT, RScA** and **RSDTA** pertain to the **Absolute Scale** | | | | | | | | | |

Table 5. Formal validation of the metrics

As a conclusion of the formal validation we have obtained that all our metrics are in the ordinal or in a superior scale. That means that they are formally valid software metrics, as remarked by [ZUS98].

## 6 Conclusion and Future Work

If we really consider that information is “the” main organizational asset, one of the primary duties of IT professionals must be assuring its quality. Information quality can be decomposed in different types of “qualities”: presentation quality, data warehouse management system quality, data quality, physical model quality and multidimensional model quality. Last one is our focus. Although some interesting guidelines have been proposed for designing “good” multidimensional models, more objective indicators are needed.

We are elaborating a set of valid metrics for measuring data warehouse quality, which can help designers in choosing the best option among more than one alternative design.

We have presented some metrics for measuring the data warehouse star design. We have applied a formal validation process to the metrics and we have obtained that all our metrics are in the ordinal or in a superior scale. That means that they are formally valid software metrics, as remarked by [ZUS98].

However, this is only the first steps in the overall metrics definition process. As we have indicated previously it is fundamental to run out empirical studies in order to prove the practical utility of the defined metrics. In this way, we are now working on the empirical validation of the metrics presented. As a result of this step (and of the complete method) we will be able to accept, discard or redefine the metrics presented in ths paper.

## References

[ADA98] C. Adamson and M. Venerable. *Data Warehouse Design Solutions*. John Wiley and Sons, 1998.

[BAS84] V.R. Basili and D. Weiss. A methodology for Collecting Valid Software Engineering Data. *IEEE Transactions on Software Engineering. SE-10. No. 6*. 728-738, 1984

[BAS99] V.R. Basili, F. Shull and F. Lanubille. Building Knowledge through families of experiments. *IEEE Transactions on Software Engineering. No. 4*. 456-473, July/August, 1999

[BOM97] M. Boman, J, Bubenko, P. Johannesson and B. Wangler. *Conceptual Modelling*, Prentice Hall, 1997

[BRI96] L.C. Briand, S. Morasca and V. Basili. Property-based software engineering measurement. *IEEE Transactions on Software Engineering.* 22(1). 68-85, 1996.

[BOU00] M. Bouzeghoub F. Fabret and H. Galhardas. Datawarehouse refreshment. Chapter 4 in *Fundamentals of Data Warehouses*. Ed. Springer, 2000.

[CAL01] C. Calero, M. Piattini and M. Genero. Metrics for controlling database complexity. Chapter III in *Developing quality complex database systems: practices, techniques and technologies*. Becker (ed), Idea Group Publishing, 2001.

[ENG96] L. English. *Information Quality Improvement: Principles, Methods and Management, Seminar, 5th Ed.*, Brentwood, TN: Information Impact International, Inc., 1996.

[FEN97] N. Fenton and S. Pfleeger. *Software Metrics: A Rigorous Aproach 2nd. Edition*. London, Chapman & Hall, 1997.

[GAR98] S.R. Gardner. Building the data warehouse, *Communications of the ACM, Vol. 41, Nr.9*. 52-60, September, 1998

[HAM96] T. Hammergren. *Data Warehousing Building the Corporate Knowledge Base*. International Thomson Computer Press, Milford, 1996.

[HAR96] V. Harinarayan, A. Rajaraman and J. D. Ullman. Implementing Data Cubes Efficiently. *Proc. of the 1996 ACM SIGMOD International Conference on Management of Data*. Jagadish, H. V. and Mumick, I. S. (eds.), 205-216, 1996.

[INM97] W. H. Inmon. *Building the Data Warehouse, second edition*, John Wiley and Sons, 1997.

[JAR00] M. Jarke, M. Lenzerini, Y. Vassiliou and P. Vassiliadis. *Fundamentals of Data Warehouses*, Ed. Springer, 2000.

[KEL97] S. Kelly. *Data Warehousing in Action*. John Wiley & Sons, 1997.

[KIM98] R. Kimball, L. Reeves, M. Ross and W. Thornthwaite. *The Data Warehouse Lifecycle Toolkit*, John Wiley and Sons, 1998.

[LAB97] W. Labio, D. Quass and B. Adelberg. Physical Database Design for Data Warehouses. *Thirteen International Conference on Data Engineering*, IEEE Computer Society, Birmingham, 277-288, 1997.

[LIN94] O. Lindland, G. Sindre and A. Solvberg. "Understanding Quality in Conceptual Modelling", *IEEE Software, Vol. 11 Nº 2*. 42-49, March 1994.

[PIA00] M. Piattini, M. Genero, C. Calero, M. Polo and F. Ruiz. *Database Quality*. *In: Advanced Database Technology and Design*. Diaz, O. and Piattini, M. (eds.) London. Artech House, 2000.

[PIA01] M. Piattini, C. Calero and M. Genero. Table oriented metrics for relational databases. Acceted for publication in *Software Quality Journal*, 2001

.

[WEY88] E.J. Weyuker. Evaluating software complexity measures. *IEEE Transactions on Software Engineering*. 14(9). 1357-1365, 1988.

[WHI97] S.A. Whitmire. *Object Oriented Design Measurement*. Ed. Wiley, 1997.

[ZUS98] H. Zuse. *A Framework of Software Measurement*. Walter de Gruyter, 1998.

**Acknowledgements**

This research is part of the CALIDAT project carried out by Cronos Ibérica (supported by the Consejería de Educación de la Comunidad de Madrid, Nr. 09/0013/1999)