Chapter IV
Interactive Quality–Oriented Data Warehouse Development

Maurizio Pighin
IS&SE-Lab, University of Udine, Italy

Lucio Ieronutti
IS&SE-Lab, University of Udine, Italy

ABSTRACT

Data Warehouses are increasingly used by commercial organizations to extract, from a huge amount of transactional data, concise information useful for supporting decision processes. However, the task of designing a data warehouse and evaluating its effectiveness is not trivial, especially in the case of large databases and in presence of redundant information. The meaning and the quality of selected attributes heavily influence the data warehouse’s effectiveness and the quality of derived decisions. Our research is focused on interactive methodologies and techniques targeted at supporting the data warehouse design and evaluation by taking into account the quality of initial data. In this chapter we propose an approach for supporting the data warehouses development and refinement, providing practical examples and demonstrating the effectiveness of our solution. Our approach is mainly based on two phases: the first one is targeted at interactively guiding the attributes selection by providing quantitative information measuring different statistical and syntactical aspects of data, while the second phase, based on a set of 3D visualizations, gives the opportunity of run-time refining taken design choices according to data examination and analysis. For experimenting proposed solutions on real data, we have developed a tool, called ELDA (Evalutation DAta warehouse quality), that has been used for supporting the data warehouse design and evaluation.
INTRODUCTION

Data Warehouses are widely used by commercial organizations to extract from a huge amount of transactional data concise information useful for supporting decision processes. For example, organization managers greatly benefit from the availability of tools and techniques targeted at deriving information on sale trends and discovering unusual accounting movements. With respect to the entire amount of data stored into the initial database (or databases, hereinafter DBs), such analysis is centered on a limited subset of attributes (i.e., datawarehouse measures and dimensions). As a result, the datawarehouse (hereinafter DW) effectiveness and the quality of related decisions is strongly influenced by the semantics of selected attributes and the quality of initial data. For example, information on customers and suppliers as well as products ordered and sold are very meaningful from data analysis point of view due to their semantics. However, the availability of information measuring and representing different aspects of data can make easier the task of selecting DW attributes, especially in presence of multiple choices (i.e., redundant information) and in the case of DBs characterized by a high number of attributes, tables and relations. Quantitative measurements allow DW engineers to better focus their attention towards the attributes characterized by the most desirable features, while qualitative data representations enable one to interactively and intuitively examine the considered data subset, allowing one to reduce the time required for the DW design and evaluation.

Our research is focused on interactive methodologies and techniques aimed at supporting the DW design and evaluation by taking into account the quality of initial data. In this chapter we propose an approach supporting the DW development and refinement, providing practical examples demonstrating the effectiveness of our solution. Proposed methodology can be effectively used (i) during the DW construction phase for driving and interactively refining the attributes selection, and (ii) at the end of the design process, to evaluate the quality of taken DW design choices.

While most solutions that have been proposed in the literature for assessing data quality are related with semantics, our goal is to propose an interactive approach focused on statistical aspects of data. The approach is mainly composed by two phases: an analytical phase based on a set of metrics measuring different data features (quantitative information), and an exploration phase based on an innovative graphical representation of DW hypercubes that allows one to navigate intuitively through the information space to better examine the quality and distribution of data (qualitative information). The interaction is one of the most important feature of our approach: the designer can incrementally define the DW measures and dimensions and both quality measurements and data representations change according to such modifications. This solution allows one to evaluate rapidly and intuitively the effects of alternative design choices. For example, the designer can immediately discover that the inclusion of an attribute negatively influences the global DW quality. If the quantitative evaluation does not convince the designer, he can explore the DW hypercubes to better understand relations among data, data distributions and behaviors.

In a real world scenario, DW engineers greatly benefit from the possibility of obtaining concise and easy-to-understand information describing the data actually stored into the DB, since they typically have a partial knowledge and vision of a specific operational DB (e.g., how an organization really uses the commercial system). Indeed, different organizations can use the same system, but each DB instantiation stores data that can be different from the point of view of distribution, correctness and reliability (e.g., an organization never fills a particular field of the form). As a result, the same DW design choices can produce different informative effects depending on the data actually stored into the DB. Then, although
the attributes selection is primarily based on data semantics, the availability of both quantitative and qualitative information on data could greatly support the DW design phase. For example, in the presence of alternative choices (valid from semantic point of view), the designer can select the attribute characterized by the most desirable syntactical and statistical features. On the other hand, the designer can decide to change his design choice if he discovers that the selected attribute is characterized by undesirable features (for instance, an high percentage of null values).

This chapter is structured as follows. First, we survey related work. Then we present the methodology we propose for supporting the DW design process, and ELDA (EvaLuation DAta warehouse quality), a tool implementing such methodology. In last section we describe the experimental evaluation we have carried out for demonstrating the effectiveness of our solution. Finally, we conclude the chapter by discussing ongoing and future works.

RELATED WORKS

In the literature, different researchers have been focused on data quality in operational systems and a number of different definitions and methodologies have been proposed, each one characterized by different quality metrics. Although Wang [1996a] and Redman [1996] proposed a wide number of metrics that have become the reference models for data quality in operational systems, in the literature most works refer only to a limited subset (e.g., accuracy, completeness, consistency and timeliness). Moreover, literature reviews e.g., [Wang et al. 1995] highlighted that there is not a general agreement on these metrics, being the concept of quality strongly context dependent. For example, timeliness has been defined by some researchers in terms of whether the data are out of date [Ballou and Pazer 1985], while other researchers use the same term for identifying the availability of output on time [Kriebel 1978][Scannapieco et al. 2004][Karr et al. 2006]. Moreover, some of the proposed metrics, called subjective metrics [Wang and Strong 1996a] e.g., interpretability and easy of understanding, require an evaluation made by questionnaires and/or interviews [Lee et al. 2001] and then result more suitable for qualitative evaluations rather than quantitative ones. Jeusfeld and colleagues [1998] adopt a meta modeling approach for linking quality measurements to different abstraction levels and user requirements, and propose a notation to formulate quality goals, queries and measurements. An interesting idea is based on detecting discrepancies among objective and subjective quality measurements [Pipino et al. 2002][De Amicis and Batini 2004].

Some researchers have been focused on methods for conceptual schema development and evaluation [Jarke et al. 1999]. Some of these approaches e.g., [Phipps and Davis 2002] include the possibility of using the user input to refine the obtained result. However, these solutions typically require to translate user requirements into a formal and complete description of a logical schema.

An alternative category of approaches employs objective measurements for assessing data quality. In this context, an interesting work has been presented in [Karr et al. 2006], where quality indicators are derived by analyzing statistical data distributions. Another interesting work based on objective measurements has been proposed in [Calero et al. 2001], where a set of metrics measuring different features of multidimensional models have been presented. However, although based on metrics that have same similarity with our proposal (number of attributes, number of keys, etc.), this solution evaluates the DW quality considering the DW schema but not the quality of initial data.

A different category of techniques for assessing data quality concerns Cooperative Information Systems (CISs). In this context, the DaQuinCIS project proposed a methodology [Scannapieco 1996b]...
et al. 2004][Missier and Batini 2003] for quality measurement and improvement. The proposed solution is primarily based on the premise that CISs are characterized by high data replication, i.e., different copies of the same data are stored by different organizations. From data quality prospective, this feature offers the opportunity of evaluating and improving data quality on the basis of comparisons among different copies. Data redundancy has been effectively used not only for identifying mistakes, but also for reconciling available copies or selecting the most appropriate ones.

From data visualization point of view, some researchers have been focused on proposing innovative solutions for DW representations. For example, Shekhar and colleagues [2001] proposed map cube, a visualization tool for spatial DWs; taken a base map, associated data tables and cartographic preferences, the proposed solution is able to automatically derive an album of maps displaying the data. The derived visualizations can be browsed using traditional DW operators, such as drill-down and roll up. However, as the need to understand and analyze information increases, the need to explore data advances beyond simple two dimensional representations; such visualizations require analysts to view several charts or spreadsheets sequentially to identify complex and multidimensional data relationships.

Advanced three dimensional representations enable analysts to explore complex, multidimensional data in one screen. In this context, several visualization and interaction techniques have been proposed, each one characterized by different functionalities, goals and purposes. The Xerox PARC User Interface Research Group has conducted an extensive research in this field, focusing on hierarchical information and proposing a set of general visualization techniques, such as perspective walls, cone trees, hyperbolic and disk trees [Robertson et al., 1993]. Different tools based on such visualization techniques have been developed; for example, in [Noser and Stucki, 2000] has been presented a web-based solution that is able to visualize and query large data hierarchies in an efficient and versatile manner starting from data stored into relational DBs. A different category of visualization techniques and tools adopt solutions that are specific to the considered application domain. A specific data visualization application is NYSE 3-D Trading Floor [Delaney, 1999], a virtual environment designed for monitoring and displaying business activities. The proposed application integrates continuous data streams from trading systems, highlights unusual business and system activities, and enables the staff to pinpoint where complex events are taking place. In the context of urban planning, an explorative work has been presented in [Coors and Jung, 1998]; the proposed tool, called GOOV1-3D, provides access and interaction with a spatial DB storing information for example on buildings. An important feature supported by the tool is the possibility to query data and observe its effects directly on data representation (e.g., the user is interested in finding buildings characterized by less than five floors). Three dimensional visualizations have been successfully employed also for representing temporal data in the medical domain, where they are used for displaying and analyzing the huge amount of data collected during medical treatments and therapies. An interesting visualization and interaction technique has been proposed in [Chittaro, Combi and Trapasso, 2003], where the specific domain of hemodialysis is considered.

For the specific context of DWs, there are not three dimensional solutions targeted at effectively supporting the DW design. More specifically, we are interested in proposing visualization and interaction techniques that are specifically devoted to highlight relations and data proprieties from data distributions point of view. Although traditional three dimensional representations can be adopted for such purposes (e.g., 3D bar charts), they do not provide the DW designer with the control needed for data examination and exploration.
PROPOSED METHODOLOGY

In this chapter we propose an interactive methodology supporting the DW design and evaluating the quality of taken design choices. Our solution is mainly based on two phases (see Figure 1): the first one is targeted at guiding the attributes selection by providing quantitative information evaluating different data features, while the second phase gives the opportunity of refining taken design choices according to qualitative information derived from the examination and exploration of data representations.

More specifically, for the first phase we define a set of metrics measuring different syntactical and statistical aspects of data (e.g., percentage of null values) and evaluating information directly derived from initial DBs (e.g., attributes types) and the current version of the DW schema (e.g., active relations). By combining obtained indexes, we derive a set of quantitative measurements highlighting the set of attributes that are more suitable to be included into the DW as dimensions and measures. According to derived information and considering data semantics, the expert can start to define a preliminary version of the DW schema (specifying the initial set of DW measures and dimensions). Given such information, for the second phase we propose interactive three-dimensional representations of DW ipercubes that allow one to visually evaluate the effects of the preliminary design choices. This solution allows one to navigate through the data intuitively, making easier the task of studying data distributions in the case of multiple dimensions, discovering undesirable data features, or to confirm selected DW measures and dimensions. If the expert catches some unexpected and undesirable data feature, he can go back to the previous phase for refining his design choice, e.g., excluding some attributes and including new dimensions and/or measures. It is important to note that each modification of the DW schema causes the indexes re-computation on the fly and the possibility of exploring different DW ipercubes. These two phases can be executed till the expert find the good compromise between his needs and the quantitative and qualitative results provided by our methodology. In the following, we describe in detail the quantitative and qualitative phases we propose for supporting the DW design process.

Quantitative Phase

Quantitative phase is based on the global indicators $M_m(t_j, a_i)$ and $M_d(t_j, a_i)$ estimating how much the attribute $a_i$ belonging to the table $t_j$ is suitable to be used respectively as DW measure and dimension. Information on the final DW design quality $M(DW)$ is then derived by considering the indicators of selected attributes. The global indicators $M_m(t_j, a_i)$ and $M_d(t_j, a_i)$ are derived by combining the indexes computed by a set of metrics, each one designed with the aim of

Figure 1. General schema of the proposed methodology
capturing a different (syntactical or statistical) aspect of data. More specifically, we differently weight each measured feature using a set of coefficients: negative coefficients are used when the measurement involves an undesirable feature for a specific point of view of the analysis (dimension or measure), while positive coefficients are used in the case of desirable features. It is important to note that in our experiments we simply use unitary values for the coefficients (i.e. –1 and 1), postponing to further evaluations the accurate tuning of these values.

The metrics we propose refer to three different DB elements:

- **Tables of a DB:** These metrics are able to measure general features of a given table, such as the percentage of numerical attributes of the table. The two indicators \( MT_m(t_j) \) and \( MT_d(t_j) \) measuring how much the table \( t_j \) is suitable to extract respectively measures and dimensions are derived by combining the indexes computed by these metrics.

- **Attributes of a table:** At a level of a single table, these metrics measure salient characteristics of data, such as the percentage of null values of an attribute. The two indicators \( MA_m(a_i) \) and \( MA_d(a_i) \) evaluating if the attribute \( a_i \) provides value added respectively as dimension and measure are derived by combining the indexes computed by these metrics.

- **Relations of a DB:** These metrics estimate the quality of DB relations. The quality indicator \( MR(t_j) \) measuring the quality of the relations involving the table \( t_j \) is used during the attributes selection for refining the indexes of the attributes belonging to the considered table. Proposed approach is interactive, since quality indicators dynamically change their value according to the measures and dimensions actually selected for the DW construction.

In this chapter, we give an informal and intuitive description of the proposed metrics; a deeper mathematical and formal description for some metrics can be found in [Pighin and Ieronutti, 2007].

### Table Metrics

In this Section, we describe the set of metrics \( mt_{e=1..k} \) (being \( k \) the total number of table metrics, in this chapter \( k = 5 \)) and corresponding indexes we propose for DB tables. With these metrics, we aim at taking into account that different tables could play different roles and then result more/less suitable for extracting measures and dimensions. The indicators \( MT_m(t_j) \) and \( MT_d(t_j) \) measuring how much the table \( t_j \) is suitable to extract measures and dimensions are derived by linearly combining the indexes computed by the metrics \( mt_{e=1..k} \) using respectively the set of coefficients \( ct_{m,e} \) and \( ct_{d,e} \). The indicators \( MT_m(t_j) \) and \( MT_d(t_j) \) are used: (i) to support the selection of the tables for the DW definition, (ii) to differently weight the indexes computed on the attributes belonging to different tables. In particular, the two indicators are derived as follows:

\[
MT_p(t_j) = \frac{\sum_{e=1}^{k} (ct_{p,e} \times mt_e(t_j))}{k}
\]

where \( p = d \) (dimension) or \( m \) (measure), \( e = 1,...,k \) identifies the metric, \( j \) identifies the table, and \( ct_{p,e} \) is the coefficient of the table-metric \( e \). In the following, we briefly describe the metrics \( mt_{e=1..5} \) we propose for DB tables and corresponding coefficients.

**Percentage of data.** This metric measures the percentage of data stored into a given table with respect to the total number of data stored into the entire DB(s). If the analysis concerns the identification of the tables that are more suitable to extract measures, the correspond-
ing coefficient is positive ($ct_{m,1} > 0$) since tables storing transactional information are generally characterized by a higher number of data with respect to the other types of tables. On the other hand, the coefficient for dimensions is negative ($ct_{d,1} < 0$) since tables concerning business objects definitions (e.g., products or clients) are typically characterized by a lower number of data than transactional archives.

**Rate attributes/records.** This metric computes the rate between the number of attributes and the number of records in the considered table. If the analysis concerns the identification of tables that are more suitable to extract dimensions, the corresponding coefficient is positive ($ct_{d,2} > 0$) since tables concerning business objects definitions are characterized by a number of attributes that is (typically lower but) comparable with the number of records stored into the table. On the other hand, the coefficient for measures is negative ($ct_{m,2} < 0$) since generally in transactional archives the number of records and the number of attributes have a different order of magnitude.

**In/out relations.** This metric measures the rate between incoming and outgoing relations. Given a one-to-many relation connecting the tables $t_{j1}$ and $t_{j2}$, in this chapter we consider the relation as incoming from the point of view of the table $t_{j2}$, while outgoing from the point of view of $t_{j1}$. If the analysis concerns the identification of tables that are more suitable to extract measures, the corresponding coefficient is positive ($ct_{m,3} > 0$) since these tables are generally characterized by an higher number of incoming relations than outgoing ones. For example, the table storing information on the bill is linked by a number of other tables storing for example information on sold products, sales agent and on the customer. For the opposite reason, the coefficient for dimensions is negative ($ct_{d,3} < 0$).

**Number of relations.** This metric considers the total number of relations involving the considered table. The computed index estimates the relevance of the table, since tables characterized by many relations typically play an important role into the DB. Since an high number of relations is a desirable feature for both measures and dimensions point of view, both coefficients are positive ($ct_{m,4}$ and $ct_{d,4} > 0$).

**Percentage of numerical attributes.** This metric derives the percentage of numerical attributes into the considered table. Integers, decimal numbers and date are considered by this metric as numerical data. Since tables storing information related with transactional activities are generally characterized by an high number of numerical attributes, the coefficient for measures is positive ($ct_{m,5} > 0$). Indeed, these tables typically contain many numerical attributes, such as ones storing information on the amount of products sold, the date of the sale, the price of different products and the total bill. On the other hand, tables storing information on products, customers and sellers are characterized by an higher number of alphanumerical attributes (e.g., specifying the customer/seller address). For this reason, if the analysis concerns the identification of the tables that are more suitable to extract dimensions, the corresponding coefficient is negative ($ct_{d,5} < 0$).

### Attribute Metrics

In this Section, we describe the set of metrics $ma_{h=1..r}$ (being $r$ the total number of attribute metrics, in this chapter $r = 6$) and corresponding indexes we propose for DB attributes. The global indicators $MA_m(a_i)$ and $MA_d(a_i)$ measuring how much the attribute $a_i$ is suitable to be used respectively as measure and dimension are derived by differently combining the indexes derived by the metrics $ma_{h=1..r}$ using respectively the set of coefficients $ca_{m,h}$ and $ca_{d,h}$. In particular, the two indicators are derived as follows:

$$\frac{\sum_{h=1}^{r}(ca_{p,h} * ma_h(a_i))}{r}$$
where \( p = d \) (dimension) or \( m \) (measure), \( h = 1, \ldots, r \) identifies the metric, \( i \) identifies the attribute, and \( ca_{p,h} \) is the coefficient of the attribute-metric \( h \) considering the role \( p \) of the attribute. In the case of a DW attribute derived as a combination of more than one DB attributes, the corresponding index is derived as the mean of the indexes related to the DB attributes. In the following, we briefly describe the metrics \( ma_{h=1,6} \) we propose for DB attributes and corresponding coefficients.

**Percentage of null values.** This metric measures the percentage of attribute data having null values. Although simple, such measurement provides an important indicator concerning the relevance of an attribute since, independently from its role, attributes characterized by a high percentage of null values are not suitable to effectively support decision processes. For example, an attribute having a percentage of null values greater than 90% is characterized by a scarce informative content from the analysis point of view. For this reason, both coefficients for this metric are negative (\( ca_{m,i} < 0 \) and \( ca_{d,i} < 0 \)), highlighting that the presence of an high number of null values is an undesirable feature for both dimensions and measures.

**Number of values.** The index computed by this metric concerns the extent in which the attribute assumes different values on the domain. More specifically, the metric behaves like a cosine function: if an attribute assumes a small number of different values (e.g., in the case of units of measurement where only a limited number of different values is admitted), the metric derives a value that is close to 1. A similar value is derived in the case of attributes characterized by a number of values that equals the total number of table records (e.g., when the attribute is the primary key of a table). Intermediate values are computed for the other cases according to the cosine behavior.

If the analysis concerns the evaluation of how much an attribute is suitable to be used as dimension, the corresponding coefficient is positive (\( ca_{d,i} > 0 \)), since both attributes assuming a limited number of different values and ones characterized by a large number of different values can be effectively used for exploring the data. For example, an attribute storing information on the payment type (e.g., cash money or credit card) is suitable to be used as dimension and typically it is characterized by limited number of different values. On the other extreme, an attributes storing information on product or customer codes is also suitable to be used as dimension and typically it is characterized by an high number of different values. With respect to the measures choice, the coefficient is negative (\( ca_{m,i} < 0 \)) because attributes characterized by (i) few values are generally not suitable to be used as measures, since they do not contain discriminatory and predictive information, and (ii) a large number of different values can correspond to keys and then result unsuitable to be used as measures. On the other hand, attributes storing information related to transactional activities (then, suitable to be used as measures) are characterized by a number of values (e.g., purchase money or number of elements sold) that is lower with respect to the total number of records.

**Degree of clusterization.** This metric measures the extent in which the attribute values are clustered on the domain. If the analysis concerns the evaluation of how much an attribute is suitable to be used as dimension, the corresponding coefficient is positive (\( ca_{d,i} > 0 \)), since attributes that are suitable to be used as dimensions (e.g., numerical codes and names of products, customers and supplier) typically are clusterizable. On the other hand, the coefficient for measures is negative (\( ca_{m,i} < 0 \)), since attributes suitable to be used as measures generally are characterized by values that tend to spread over the domain. It is important to highlight that this metric does not consider the data distribution into clusters, but only the total number of clusters into the attribute domain.
**Uniformity of distribution.** This metric measures how much the values of an attribute are equally distributed on the domain. The possibility of highlighting uniform distributions enables our methodology to identify attributes that are suitable to be used as measures, since typically they are not characterized by uniform distributions (e.g., normal distribution). For example, it is more probable that the distribution of values of an attribute storing information on the customer is more similar to an uniform distribution with respect to the distribution of an attribute storing information on the bill (typically characterized by a Gaussian distribution).

For this reason, if the analysis concerns the evaluation of how much an attribute is suitable to be used as a measure, the corresponding coefficient is negative \( (ca_{m,4} < 0) \). On the other hand, if the analysis concerns dimensions, the corresponding coefficient is positive \( (ca_{d,4} > 0) \); indeed, the more values are uniformly distributed on the domain (or in the considered subset), the more effectively the analyst can explore the data.

**Keys.** This metric derives a value both taking into account if the considered attribute belong or not to primary and/or duplicable keys. The coefficient for dimensions is positive \( (ca_{d,5} > 0) \) since attributes belonging to the primary or secondary keys often identify look-up tables and then they are the best candidates for the DW dimensions. On the other hand, the coefficient for measures is negative \( (ca_{m,5} < 0) \) since attributes belonging to primary or secondary keys typically are not suitable to be used as measures.

**Type of attribute.** This metric returns a float value according to the type of the attribute (alphanumerical strings = 0, whole numbers or temporal data = 0.5, real numbers = 0). Typically numerical attributes are more suitable to be used as measures rather than being used as dimensions; for this reason, the coefficient for measures is positive \( (ca_{m,6} > 0) \). On the other hand, in the case of dimensions, the corresponding coefficient is negative \( (ca_{d,6} < 0) \) since business objects definitions are often coded by alphanumerical attributes. Moreover, alphanumerical attributes are rarely use in a DW as measures due to the limited number of applicable mathematical functions (e.g., count function).

**Relation Metrics**

In this Section, we describe the set of metrics \( MR_{s=1..f} \) (being \( f \) the total number of relation metrics, in this chapter \( f = 2 \)) and corresponding indexes we propose for DB relations. These metrics have been designed with the aim of measuring the quality of relations by considering (i) data actually stored into the DB and (ii) relations actually used into the DW. Information on relations quality is used during the DW construction to dynamically refine the indexes referring to DB attributes and tables. As a result, unlike table and attribute indexes that are computed only once on the initial DB, these indexes are updated whenever the DW schema changes (e.g., new measures are included into the DW). This solution allows the methodology to consider the quality of the relations that are actually used for exploring the data into the DW, enabling to (i) better support the user during the selection of measures and dimensions, and (ii) estimate more precisely the final DW design quality.

For the evaluation, we define \( MR(a_{i_1}, a_{i_2}) \) as a quality indicator for the relation directly connecting the attributes \( a_{i_1} \) and \( a_{i_2} \).

In particular, such indicator is derived by combining the indexes computed by the metrics \( MR_{s=1..f} \) as follows:

\[
MR(a_{i_1}, a_{i_2}) = \frac{\sum_{s=1}^{f} MR_{s}(a_{i_1}, a_{i_2})}{f}
\]

where \( s = 1, ..., f \) identifies the metric, while \( a_{i_1} \) and \( a_{i_2} \) the DB attributes connected by a direct relation. Once these indicators are computed, our methodology derives for each table \( t_j \) the indicator
Data Warehouse Quality Metric

Our methodology derives for each attribute \( a_i \) belonging to the table \( t_j \) the two global indicators \( M_{\text{d}}(t_j, a_i) \) and \( M_{\text{m}}(t_j, a_i) \) indicating how much the attribute is suitable to be used in the DW respectively as dimension and measure. These indicators are computed by combining the attribute, table and relation indexes described in previous sections. More specifically, these indicators are derived as follows:

\[
M_{\text{d}}(t_j, a_i) = MT_p(t_j) \cdot MA_p(a_i) \cdot MR_p(t_j) \quad a_i \in t_j
\]

where \( p = d \) (dimensions) or \( m \) (measure), \( i \) and \( j \) identify respectively the considered attribute and table, \( MT_p, MA_p \) and \( MR_p \) are respectively the table, attribute and relation indexes.

Once all indicators are computed, our methodology derives two ordered lists of DB attributes: the first list contains the attributes ordered according to \( M_n \) while the second one according to \( M_d \). The two functions \( \text{rank}_m(a) \) and \( \text{rank}_d(a) \) derive the relative position of the attribute \( a_i \) respectively into the first and second (ordered) list. It is important to note that while \( M_n \) and \( M_d \) are used for deriving information concerning the absolute quality of an attribute, \( \text{rank}_m(a) \) and \( \text{rank}_d(a) \) can be used for evaluating the quality of an attribute with respect to the quality of the other DB attributes.

Finally, let \( D_{\text{dw}} \) be the set of \( n_d \) attributes chosen as DW dimensions and \( M_{\text{dw}} \) the set of \( n_m \) attributes selected as measures, the final DW design quality \( M(DW) \) is estimated as follows:

\[
M(DW) = \frac{\sum_{a_i \in D_{\text{dw}}} M_m(t_j, a_i) + \sum_{a_i \in D_{\text{dw}}} M_d(t_j, a_i)}{n_m + n_d}
\]
Qualitative Phase

The qualitative phase is based on an interactive three-dimensional representation of DW ipercubes that allows one to better evaluate data distributions and relations among attributes selected in the previous phase. In particular, each DW ipercube (characterized by an arbitrary number of different dimensions and measures) can be analyzed by exploring and studying different sub-cubes, each one characterized by three dimensions and one measure. Each dimension of the representation corresponds to a descriptive attribute (i.e., dimension), while each point into the three-dimensional space corresponds to a numeric field (i.e., measure). At any time the user can change the considered measure and dimensions and the ipercube representation changes according to such selection.

Since representing each fact as a point into the ipercube space can be visually confusing (e.g., millions of records are represented as millions of points overlapping each other), we propose to simplify the representation by discretizing the three-dimensional space and using different functions for grouping facts falling into the same discretized volume, represented by a small cube. The user can interactively change the granularity of the representation by modifying the level of discretization (consequently, the cubes resolution) according to his needs. Small cubes are more suitable for accurate and precise analysis, while a lower level of discretization is more suitable whenever it is not required an high level of detail (e.g., for providing an overview of data distribution).

In general, the user could select both a particular grouping function and the granularity of the representation according to the purposes and goals of the analysis. The main grouping functions are count, sum, average, standard deviation, minimum and maximum value. For example, the user could select the count function for studying the distribution of products sold to different clients during the last year. In particular, both representations depicted in Figure 2 refer to such kind of data, but they differ from the point of view of representation granularity.

Additionally, we propose a set of interaction techniques for allowing the user to intuitively explore the DW ipercubes. More specifically, we suggest the use of the color coding, slice and dice, cutting plane, detail-on-demand and dynamic queries techniques for enabling the user to analyze

Figure 2. Representing the same data using (a) 8 and (b) 24 discretizations
Interactive Quality-Oriented Data Warehouse Development

the ipercubes, visual representations that can also be examined using multiple point of view. We separately present and discuss in more detail the above techniques in the following subsections.

It is important to note that the real-time interaction is achieved into the visualization since most proposed interaction techniques work on aggregated data (i.e., a discretized version of the initial ipercube); such solution allows one to reduce considerably the time for performing the required computations.

Color Coding

In the proposed representation, a color coding mechanism is used for mapping numerical data to visual representations. This solution allows one to intuitively evaluate data distributions and easily identify the outliers, avoiding to examine and interpret numerical values [Schulze-Wollgast et al., 2005]. The proposed mapping allows the user to:

- Choose between using two or more control points (each one associating a value with a color) and select appropriate control points,
- Fine-tune color coding, by controlling the transitions between colors. In particular, the user can set the parameter used for exponential interpolation between colors. With respect to linear interpolation, this solution allows one to more clearly highlight subtle differences between values [Schulze-Wollgast et al., 2005] and is particularly effective when values are not uniformly distributed (as it often happens in our case).

The color coding mechanism employed for both representations depicted in Figure 2 is based on two control points and a simple linear interpolation between the two colors: the cyan color is used for representing the minimum value, while the red color is used for coding the maximum value.

It is important to note that the user can interactively modify the exponent of the interpolation and the visualization changes in real-time according to such modification. This functionality provides the user with the possibility of intuitively and interactively exploring numeric data from a qualitative point of view. In this chapter, most figures refer to representations based on a color coding mechanism characterized by two control points (cyan and red respectively for the minimum and maximum value).

Slice and Dice

In the context of DWs, slice and dice are the operations that allow one to break down the information space into smaller parts to better focus the data examination and analysis on specific dimensions ranges. In the proposed interaction technique, the selection of data subset is performed through rangesliders (i.e., graphical widget that allows one to select an interval of values), each one associated to a different ipercube dimension. By interacting with the rangesliders, the user has the possibility to select proper ranges of domain values and the visualization changes in real-time according to such selection. For example, Figure 3 (a) depicts the initial ipercube where all facts are considered and represented into the corresponding visualization. In Figure 3 (b) only a subset of data has been selected and visualized; more specifically, only records satisfying the logic formula \((\text{CARTE} < \text{product} < \text{INDURITORE}) \text{ AND } (\text{CROAZIA} < \text{broker} < \text{GRECIA}) \text{ AND } (2002-10-21 < \text{sold date} < 2003-01-24)\) are considered and represented. Conceptually, each rangeslider controls a couple of cutting planes corresponding respectively to the upper and lower bounds of a domain subinterval; only data located between such planes are represented.

Once the user has selected the appropriated part of the ipercube space, the dice (or slice) operations can be performed on such data; this operation allows one to obtain a more detailed
Interactive Quality-Oriented Data Warehouse Development

Figure 3. Using rangesliders for selecting a specific subset of ipercube data: (a) the whole set of records is considered, (b) different ranges of domain values are considered and (c) a dice operation is performed on the specified data subset

visualization concerning the selected ipercube space, as depicted in Figure 3 (c). It is important to note that such operation can be performed time after time to incrementally increase the level of detail of the visualization, giving at each step the opportunity of identifying the most interesting part of the information space.

Cutting Plane

In computer graphics, the term occlusion is used to describe the situation in which an object closer to the viewpoint masks a geometry further away from the viewpoint. This problem can considerably affect the effectiveness of three dimensional visualizations, especially in the case of representations characterized by a high number of different objects and geometries. There are mainly two solutions for overcoming the occlusion problem. The first one is based on the usage of semitransparent (i.e., partially transparent) objects. For example, it has been demonstrated that such solution has positive effects on navigation performance (Chittaro and Scagnetto, 2001). Unfortunately, this solution can not be effectively applied together with color coding mechanisms, since modifications to the degree of transparency of an object heavily influence the colors perception for both close (semitransparent) and distant (solid) geometries.

Another solution is based on the usage of cutting planes, virtual planes that are used for partitioning the three dimensional space in two parts; only objects belonging to one partition are displayed into the visualization, while the other ones are hidden (they become completely transparent).

In the proposed methodology, a cutting plane can be used for exploring the ipercube in the case of dense data. The user can interactively modify the vertical position and the orientation of the cutting plane, allowing one to examine internal parts of the ipercube. Figure 4 demonstrates the benefits of such solution. In particular, Figure 4 (a) depicts an ipercube characterized by dense data; from such representation, the user cannot derive any information concerning the internal data, since only data belonging to the ipercube surfaces is visible. By modifying the rotation and position of the cutting plane the user can
Figure 4. Exploring the ipercube representation using the cutting plane technique

(a) (b) (c)

easily explore the entire ipercube, discovering for example that the minimum value is positioned at the centre of the ipercube, as depicted in Figure 4 (b) and (c), where two different rotations are used for exploring the data.

**Detail-On-Demand**

In the proposed representation, we suggest also the use of the *detail-on-demand* method: starting from a data overview, the user has the possibility of obtaining detailed information referring to a particular part of data without losing sight of the ipercube overview. Then, instead of incrementally refining the analysis using slice and dice operations, the detail-on-demand technique allows the user to go deep into data, enabling to access information at the lowest level of detail. In particular, as soon as the user selects a specific part of the ipercube representation, both textual and numerical information on records corresponding to the selection is retrieved and visualized.

**Dynamic Queries**

*Dynamic queries* [Shneiderman, 1994] are a data analysis technique that is typically used for exploring a large dataset, providing users with a fast and easy-to-use method to specify queries and visually present their result. The basic idea of dynamic queries is to combine input widget, called *query devices*, with graphical representation of the result. By directly manipulating query devices, user can specify the desired values for the attributes of elements in the dataset and can thus easily explore different subset of data. Results are rapidly updated, enabling users to quickly learn interesting proprieties of data. Such solution has been successfully employed in different application domains, such as real estate [Williamson and Shneiderman, 1992] and tourism [Burigat et al., 2005].

We adopt the dynamic queries technique for making easier the task of identifying the subset of data satisfying particular measure proprieties through a set of rangesliders, the same graphical widget used for performing slice and dice operations (see Section “Slice and Dice”). Instead of acting on dimensions ranges, in this case the rangesliders are used for querying the values concerning different grouping functions. More specifically, by interacting with such graphical widget, the user can modify the range of values for a specific grouping function and the visualization is updated in real-time; as a result, only data satisfying all conditions are displayed into the representation. Supported conditions refer to available grouping functions, i.e. count, average, sum, maximum,
minimum and standard deviation. This solution allows one to easily highlight data (if exist) characterized by particular proprieties (e.g., outliers). For example, in Figure 5 we consider the ipercube characterized by “broker”, “sold date” and “product class” as dimensions and “product quantity” as measure. By interacting with two rangesliders (one for each constraint), the user has the possibility to easily identify the information spaces characterized by more than a certain number of records (see Figure 5 (a) and (b)) and where the total number of products sold is less than a given threshold (see Figure 5 (c)). It is important to note that all representations depicted in Figure 5 refer to the counting function (highlighted in green). As a result, the color of each cube codes the number of records falling in the corresponding discretized volume; any time the user can decide to change such choice by simply selecting a different grouping function (e.g., sum function) and the color mapping of the representation changes according to such selection.

Viewpoint Control

In traditional tabular data representations, the pivot allows one to turn the data (e.g., swapping rows and columns) for viewing it from different perspectives. However, in the case of two dimensional representations the available possibilities are very limited. One of the most important feature of three dimensional data representations is the possibility of observing the content from different points of view (called viewpoints). This way, users can gain a deeper understanding of the subject and create more complete and correct mental models to represent it [Chittaro and Ranon, 2007]. For example, different viewpoints can be designed with the purpose of focusing the user attention towards different data aspects or with the aim of highlighting particular relations among data. Indeed, the benefits provided by the availability of alternative viewpoints have been successfully exploited by several authors (e.g., [Li et al., 2000][Campbell et al., 2002]) for proposing three dimensional representations characterized by effective inspection possibilities.

The ipercube representation can be explored by using three different viewpoint categories: free, fixed and constrained. The first category allows the user to completely control the orientation of the viewpoint; with such control, the user has the possibility of freely positioning and orienting the point of view to better observe a particular

Figure 5. Querying and visualizing data ipercube
aspect of the representation. However, such freedom can introduce exploration difficulties, especially in the case of users that are not expert in navigating through three dimensional spaces. In this situation, the effort spent in controlling the viewpoint overcomes the benefits offered by such navigation freedom.

In the second category the viewpoint position and orientation is pre-determined; the user can explore the representation from different points of view by simply changing the currently selected viewpoint. For such purpose, we suggest eight viewpoints (one for each ipercube vertex) that provide the users with meaningful axonometric views of the ipercube representation (see Figure 6).

The last category of viewpoints is the more interesting from data exploration and analysis point of view. Each viewpoint is constrained to a different three dimensional surface, meaning that it is positioned to a fixed distance with respect to the surface and oriented perpendicularly with respect to the surface. If the surface changes its position and/or orientation, the corresponding viewpoint position and orientation are updated according to the constraints.

We proposed seven constrained viewpoints, one constrained to the cutting plane (see Section “Cutting Plane”) and the remaining viewpoints constrained to the six dice planes (see Section “Sele and Dice”). As a result, each constrained viewpoint is able to focus the user attention toward parts of the representation involved in the current interaction, simplifying at the same time the complexity of the visualization and reducing the effort required for controlling the viewpoint.

ELDA TOOL

ELDA (EvaLuation DATawarehouse quality) is a tool designed for experimenting proposed methodology on real data. It has been developed by carefully taking into account human-computer interaction issues and focusing on computation performance. The task of evaluating a DW using ELDA is mainly composed by two phases, separately described in the following sections.

Quantitative Phase

In ELDA the quantitative phase is composed by two sequential steps. In the first step ELDA (i)
computes table and attribute indexes (see Sections “Table metrics” and “Attribute metrics”), and (ii) measures the main features of direct relations (see Section “Relation metrics”). The time required for such measurements strictly depends on the number of records stored into the DB(s). For example, in our experiments (involving tens of tables, hundreds of attributes and millions of records) the computation takes about ten minutes on a Pentium 4 2GHz processor with 1Gb ram.

Once all indexes are computed, in the second step ELDA combines them with the corresponding coefficients to derive (i) for each DB table $t_j$ the global indicators $MT_d(t_j)$ and $MT_m(t_j)$, and (ii) for each DB attribute $a_i$ belonging to the table $t_j$ the global indicators $M_d(t_j, a_i)$ and $M_m(t_j, a_i)$. More specifically, according to the current role of the analysis (i.e. dimension or measure), the tool ranks and visualizes into two ordered lists the corresponding indicators, as depicted in the lower part of Figure 7. As a result, tables and attributes that are more suitable for the selected role are positioned in the first rows of the lists. In addition to quality measurements specified in the last column of lists depicted in Figure 7, ELDA also provides information both on the absolute (first column) and relative (second column) position of the considered DB element (table and attribute) into the corresponding ranked list. As soon as the user changes the point of view of the analysis (e.g., from dimension to measure), the tool updates the ranked lists according to the current user choice.

An important functionality offered by ELDA is the possibility of filtering the list of DB attributes. In particular, the tool visualizes only the attributes belonging to the tables that have been selected into the tables list. This functionality is particularly effective in the case of DBs characterized by an high number of tables and attributes; in such situations, the user can start the analysis only by considering the attributes belonging to high-ranked tables, and then extend the analysis to the other attributes.

Ranked and filtered attributes list can be effectively used for supporting the selection of DW measures and dimensions, since is a concise but effective way for providing users with statistical and syntactical information. According to semantic considerations and guided by computed quality indicators, the user can start to include dimensions and measures by directly clicking on the corresponding rows of the list. As a result, selected attributes are added to the list of DW measures.
Interactive Quality-Oriented Data Warehouse Development

or dimensions depending on the current role of the analysis; beside the name of the attributes, ELDA also includes information concerning the computed quality measurements. It is important to note that each DW schema modification can cause the inclusion/exclusion of (direct or indirect) relations connecting measures and dimensions. Every time such situation occurs, ELDA (i) recomputes proper relation indexes (using pre-computed information on direct relations) and (ii) consequently refines at selection-time both tables and attributes indicators, ranking both lists according to new measurements.

The following two additional functionalities have been designed with the aim of making easier and more intuitive the task of evaluating taken DW design choices. The first functionality consists in counting the number of selected DW measures and dimensions falling into different rank intervals. For such evaluation, ELDA subdivides the rank values into six intervals; the more the number of attributes fall into the first intervals, the more taken choices are evaluated by the tool as appropriate. In the example depicted in Figure 7, the selected measure falls into the second interval, while dimensions fall respectively into the second, third and fifth intervals.

The second functionality offered by the tool is the possibility to visually represent the quality of taken DW design choices. For such purpose, ELDA uses a coordinate system where the x-axis is used for ranks while the y-axis for quality indicators. In the visualization small points are used for representing unselected DB attributes, while the symbol X is used for identifying DW measures and dimensions (see Figure 8). In addition to evaluate the rank of the attributes, this representation also allows one to analyze the trend of quality indicators (e.g., the user can discover sudden falls).

**Qualitative Phase**

The second phase concerns the qualitative evaluation of taken design choices. At any time the user can require to visualize a particular DW hypercube by selecting the corresponding attributes choosing among the dimensions and measures included into the current version of the DW schema. This way, the user has the possibility to constantly verify the correctness of his choices, without requiring to concretely build the entire DW for discovering unexpected and undesirable data features.

For such qualitative analysis, ELDA provides the user with several controls and functionalities that allow one to interactively explore and examine data representations. More specifically, at the beginning the user has to specify the set of dimensions, the measure and the related grouping function to be used for the analysis. Moreover, the user has also the possibility of selecting the proper granularity of the representation taking into account two factors. First, the choice is influenced by the required resolution of the representation. For example, while high-resolution representations are more suitable for accurate and precise analysis, lower resolutions are more suitable for providing the user with a data distribution overview. Second, since the granularity influences the time required for the computation, the choice also depends on the available processing power. For example, in our experiments (involving tens
of tables, hundreds of attributes and millions of records, performed on a Pentium 4 2GHz processor with 1Gb ram) the ipercube computation takes about 2 and 20 seconds using respectively 10 and 30 discretizations. However, once the representation is computed and visualized, the interaction and exploration techniques discussed in previous sections (es., cutting plane, dynamic queries and viewpoint control) can be executed in real-time since performed on aggregated data.

According to user selections, ELDA computes and visualizes the corresponding three dimensional ipercube representation on the right part of the graphical user interface (see right part of Figure 9). Then, the user has the possibility of exploring and navigating through the data by interacting with several controls (see left part of Figure 9) and the visualization changes in real-time according to such interactions. In the example depicted in Figure 9, a specific subpart of the dimensions space is considered and visualized (specifying the range of values for the attributes sold date and broker, see the top-left part of the figure) and a particular fixed viewpoint is selected for observing the representation.

The user has the possibility to gradually focus the analysis on a specific part of the ipercube using slice and dice operations, or directly obtain detailed information concerning a particular part of the ipercube by simply selecting the corresponding cube into the representation. More specifically, as soon as the mouse pointer is over a specific part of the visualization, information on dimensions concerning the (implicitly) selected space appears on the screen, as depicted in Figure 10 (a). If the user is interested in studying in more detail data falling into such space, he has simply to click the corresponding volume; a detailed report including information on all records falling into the selected volume is then displayed. Such report is displayed into a separate windows that also includes information on the grouping functions referred to the selected subset of data, as depicted in Figure 10 (b).

At any time of the evaluation, the user can change the color coding mechanism for highlighting subtle differences between data values. For such purpose, the user can choose a proper number of control points, the color associated to each control point, and the exponent used for the interpolation. For example, in the representations depicted in Figure 11, three different coding are employed for representing the same ipercube.
Interactive Quality-Oriented Data Warehouse Development

(a) and (b) differ in both the number of control points and the color associated to each control point, Figure 11 (b) and (c) differ only in the exponent used for the interpolation.

More specifically, the color coding employed for the ipercube represented in Figure 11(a) is characterized by three control points: the yellow color is employed for the minimum value, cyan color for the value located at the middle of values range, while purple color is used for the maximum value. A linear interpolation is used for mapping intermediate colors. On the other hand, two control points characterize the coding of both representations depicted in Figure 11 (b) and (c); the two coding differ only from the point of view of color interpolation. In Figure 11(a) a linear interpolation is used for the mapping, while the representation depicted in Figure 11(b) employs an exponential interpolation (in this case, the exponent equals 4) for deriving intermediate colors. In the considered examples, the latter representation is able to highlight more clearly subtle differences into the values.

If during the data exploration the user discover some unexpected and undesirable data features, he can go back to the previous phase to refine

Figure 11. The same ipercube represented using three different color coding

(a) (b) (c)
her design choices, e.g., excluding some DW dimensions and measures. It is interesting to note that although designed mainly for supporting the evaluation of data distributions, the visualization and interaction techniques proposed for the qualitative analysis allow one also to perform some preliminary data analysis, e.g., intuitive identification of the most sold products, interesting customers and productive brokers.

**EXPERIMENTAL EVALUATION**

We have experimented proposed methodology on three DBs subsets of two real world ERP (Enterprise Resource Planning) systems. Considered DBs, called respectively $DB01$, $DB02$ and $DB03$, are characterized by tens of tables, hundreds of attributes and millions of records. In particular, while $DB01$ and $DB02$ correspond to different instantiations of the same DB schema (it is the same business system used by two different commercial organizations), $DB03$ has a different DB schema (it is based on a different ERP system).

For the experimental evaluation, we asked to an expert to build an unique (and relatively simple) schema for a selling DW by selecting the attributes that are the most suitable to support decision processes. The DW build by the expert is characterized by a star schema where six attributes are used as measures ($n_m = 6$) and nine as dimensions ($n_d = 9$). Starting from this schema, we build three DWs, filling them with the three different DB sources. As a result, the attributes chosen to build the first two DWs are physically the same (since they belong to the same DB schema), while a different set of attributes (characterized by the same semantics with respect to ones selected for previous DWs) are chosen for the DW03 construction.

Then, we have experimented our methodology for testing its effectiveness by considering the above three case studies. The analysis is mainly targeted at evaluating if the proposed metrics effectively support quantitative analysis by taking into account (i) the structure of the initial DB (in this experiment, two different DB schemas are considered), (ii) data actually stored into the initial DB (in this experiment, three different data sources are considered), and (iii) the DW schema (in this experiment, an unique DW schema is considered). We have then evaluated both if during the DW construction the proposed methodology effectively drives design choices and, at the end of the quantitative phase, if it can be used for deriving information on the final DW design quality.

**Quantitative Phase**

In the first phase of our experiment, we have considered the metrics we propose for the DB tables and evaluated their effectiveness in highlighting tables that are suitable to be used for extracting measures and dimensions. The global indexes $MT_d$ and $MT_m$ for the three DBs are summarized respectively in Table 1 (a) and (b). Derived quality measurements for the DB tables are consistent with our expectations; for example, for both $DB01$ and $DB02$, the procedure highlights that $xsr$ and $intf$ are tables suitable for extracting measures since these tables store selling and pricing information. It is interesting to note that although based on the same DB schema, different indexes are computed for $DB01$ and $DB02$ due to different data distributions. A similar good result is obtained for $DB03$, where the tables $bolla_riga$ and $bolla_riga_add$ store the same kind of information stored into $xsr$, while $mag_costo$ stores pricing information on products. With respect to dimensions choice, our procedure highlights both in $DB01$ and $DB02$ the tables $gum$ and $zon$; indeed, the first table stores information on customers categories, while the second one stores geographical information on customers. A similar result is obtained for $DB03$, since the tables $anagrafico_conti$ and $gruppo_imprend$ store information respectively on customer accounts and product categories.
Interactive Quality-Oriented Data Warehouse Development

Table 1. List of DB01, DB02 and DB03 tables ranked according to (a) $MT_d$ and (b) $MT_m$

<table>
<thead>
<tr>
<th>Tables of DB01</th>
<th>$MT_d$</th>
<th>Tables of DB01</th>
<th>$MT_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>zon</td>
<td>0.8814</td>
<td>xsr</td>
<td>0.8521</td>
</tr>
<tr>
<td>gum</td>
<td>0.8645</td>
<td>org</td>
<td>0.8420</td>
</tr>
<tr>
<td>smag</td>
<td>0.8514</td>
<td>intf</td>
<td>0.8340</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tables of DB02</th>
<th>$MT_d$</th>
<th>Tables of DB02</th>
<th>$MT_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>gum</td>
<td>0.7481</td>
<td>xsr</td>
<td>0.8316</td>
</tr>
<tr>
<td>zon</td>
<td>0.7463</td>
<td>intf</td>
<td>0.8276</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tables of DB03</th>
<th>$MT_d$</th>
<th>Tables of DB03</th>
<th>$MT_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ord_tipo</td>
<td>0.8716</td>
<td>bolla_riga</td>
<td>0.8468</td>
</tr>
<tr>
<td>anagrafico_conti</td>
<td>0.8689</td>
<td>bolla_riga_add</td>
<td>0.8462</td>
</tr>
<tr>
<td>gruppo_imprend</td>
<td>0.8660</td>
<td>mag_costo</td>
<td>0.8333</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

In the second phase of the experiment, we have considered the metrics we propose for the attributes. We summarize in Table 2 (a) and (b) the quality indicators respectively from dimensions and measures point of view. The computed indexes are consistent with our expectations; for example, in both DB01 and DB02 the attributes $zn\_sigla$ and $mp\_sigla$ result suitable to be used as dimensions; indeed, the first attribute stores geographical information on customers and sellers, while the second one collects information on payment types. Additionally, our procedure identifies $lio\_prezzo$ and $xr\_valore$ as the attributes that are more suitable to be used as measures in DB01. This is consistent with the semantics of data, since the first attribute stores pricing information on special offers, while the second one refers to invoices amounts. Also in the case of DB02 the procedure highlights attributes storing money-related information; for example, the attribute $mv\_imp\_val$ stores information on accounts movements. A good result is also obtained for DB03; in this case, the procedure correctly identifies $tipo\_ord$ and $cod\_moneta$ as attributes suitable to be used as dimensions and less effective as measures. Indeed, these attributes store information respectively on types of orders and moneys. On the other hand, the attribute $qta\_ordinata$ storing information on the number of products ordered by the customer, it results suitable to be used as measure.

In the third phase of our experiment, we have considered the DW built by the expert and analyzed the rank of selected attributes in order to evaluate the effectiveness of our methodology in correctly measuring the quality of the attributes according to their role into the DW. In Table 3 and Table 4 we report respectively the measures and dimensions chosen for building the three DWs and related ranks.

To better evaluate the results, we illustrate in Figure 12 the whole set of DB attributes ranked according to $M_a$ and $M_p$, highlighting the measures and dimensions chosen by the expert to built the DW. It is interesting to note that most selected attributes (in the figure, represented by red X) are located in the upper-left part of the figures, meaning that the derived quality indicators are consistent with the expert design choices.

The final step of the quantitative phase concerns the evaluation of the derived global indicators measuring the quality of the considered DWs. From computed measurements, DW01 results the better DW, while DW02 result the worst one, due to both the low quality of data stored into the selected DB attributes and the initial DB schema. In particular, the following global indicators are computed: $M(DW01) = 0.8826$, $M(DW02) = 0.6292$ and $M(DW03) = 0.8504$. 
### Table 2. Attributes of DB01, DB02 and DB03 ranked according to (a) $MA_d$ and (b) $MA_m$

(a) Dimensions

<table>
<thead>
<tr>
<th>Attributes of DB01</th>
<th>$MA_d$</th>
<th>rank$_d$</th>
<th>Attributes of DB02</th>
<th>$MA_d$</th>
<th>rank$_d$</th>
<th>Attributes of DB03</th>
<th>$MA_d$</th>
<th>rank$_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mp_sigla</td>
<td>0.7268</td>
<td>0.0000</td>
<td>zn_sigla</td>
<td>0.6843</td>
<td>0.0024</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

(b) Measures

<table>
<thead>
<tr>
<th>Attributes of DB01</th>
<th>$MA_m$</th>
<th>rank$_m$</th>
<th>Attributes of DB02</th>
<th>$MA_m$</th>
<th>rank$_m$</th>
<th>Attributes of DB03</th>
<th>$MA_m$</th>
<th>rank$_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>lio_prezzo</td>
<td>0.7467</td>
<td>0.0000</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>xr_valore</td>
<td>0.7443</td>
<td>0.0024</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### Table 3. Ranking of DW01, DW02 and DW03 measures

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>$M_n$</th>
<th>rank$_n$</th>
<th>$M_n$</th>
<th>rank$_n$</th>
<th>$M_n$</th>
<th>rank$_n$</th>
<th>$M_n$</th>
<th>rank$_n$</th>
<th>$M_n$</th>
<th>rank$_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>product quantity</td>
<td>xr_qta</td>
<td>1.0369</td>
<td>0.9291</td>
<td>0.9616</td>
<td>0.0123</td>
<td>0.0193</td>
<td>0.0081</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>product price</td>
<td>xr_valore</td>
<td>1.2145</td>
<td>1.1629</td>
<td>0.7925</td>
<td>0.0000</td>
<td>0.0044</td>
<td>0.1071</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>broker commission</td>
<td>xr_prov_age</td>
<td>0.9999</td>
<td>0.0000</td>
<td>0.8164</td>
<td>0.0197</td>
<td>1.0000</td>
<td>0.0727</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>customer discount</td>
<td>xr_val_sco</td>
<td>0.9608</td>
<td>0.0000</td>
<td>0.8914</td>
<td>0.0468</td>
<td>1.0000</td>
<td>0.0222</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>product last cost</td>
<td>a_ult_prz_pag</td>
<td>1.0477</td>
<td>0.8452</td>
<td>0.9255</td>
<td>0.0074</td>
<td>0.0400</td>
<td>0.0121</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>product std. cost</td>
<td>a_prz_pag_stand</td>
<td>0.0339</td>
<td>0.8758</td>
<td>0.9255</td>
<td>0.9634</td>
<td>0.0267</td>
<td>0.0121</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 4. Ranking of DW01, DW02 and DW03 dimensions

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>$M_d$</th>
<th>Rank$_d$</th>
<th>$M_d$</th>
<th>Rank$_d$</th>
<th>$M_d$</th>
<th>Rank$_d$</th>
<th>$M_d$</th>
<th>Rank$_d$</th>
<th>$M_d$</th>
<th>Rank$_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>product</td>
<td>a_sigla_art</td>
<td>cod_articolo</td>
<td>1.0648</td>
<td>1.0712</td>
<td>1.0128</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0060</td>
<td></td>
<td></td>
</tr>
<tr>
<td>product class</td>
<td>smg_tipo_codice</td>
<td>cod_ricl_ind_ricl_f1</td>
<td>0.7906</td>
<td>0.7803</td>
<td>0.7092</td>
<td>0.0343</td>
<td>0.0133</td>
<td>0.0986</td>
<td></td>
<td></td>
</tr>
<tr>
<td>warehouse class</td>
<td>a_cl_inv</td>
<td>cod_ricl_ind_ricl_f2</td>
<td>0.6098</td>
<td>0.0000</td>
<td>0.7092</td>
<td>0.1397</td>
<td>1.0000</td>
<td>0.0986</td>
<td></td>
<td></td>
</tr>
<tr>
<td>customer</td>
<td>se_cod_s_conto</td>
<td>conti_clienti_m_p</td>
<td>0.8094</td>
<td>0.7497</td>
<td>0.7977</td>
<td>0.0294</td>
<td>0.0192</td>
<td>0.0523</td>
<td></td>
<td></td>
</tr>
<tr>
<td>customer class</td>
<td>gu_codice</td>
<td>cod_gruppo</td>
<td>0.6789</td>
<td>0.7479</td>
<td>0.9381</td>
<td>0.0833</td>
<td>0.0222</td>
<td>0.0141</td>
<td></td>
<td></td>
</tr>
<tr>
<td>province</td>
<td>xi_prov</td>
<td>cod_provincia</td>
<td>0.5576</td>
<td>0.7009</td>
<td>0.9482</td>
<td>0.1961</td>
<td>0.0385</td>
<td>0.0101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>country</td>
<td>ps_sigla_paese</td>
<td>elenco_stati_cod_iso</td>
<td>0.7770</td>
<td>0.8403</td>
<td>0.8044</td>
<td>0.0368</td>
<td>0.0074</td>
<td>0.0483</td>
<td></td>
<td></td>
</tr>
<tr>
<td>broker</td>
<td>ag_cod_agente</td>
<td>conti_fornitori_m_p</td>
<td>0.9302</td>
<td>0.0000</td>
<td>0.6179</td>
<td>0.0025</td>
<td>1.0000</td>
<td>0.0986</td>
<td></td>
<td></td>
</tr>
<tr>
<td>commercial zone</td>
<td>zn_sigla</td>
<td>cod_zona_comm</td>
<td>0.7977</td>
<td>0.7348</td>
<td>0.9070</td>
<td>0.0368</td>
<td>0.0266</td>
<td>0.0201</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Qualitative Phase

At the end of the quantitative phase, we have used the ELDA tool to visually analyze taken design choices for better evaluating if data distributions are coherent with the designer expectations or are characterized by some unexpected and undesirable behavior. For such purpose, we have considered, visualized and analyzed different ipercubes using possible combinations of selected DW dimensions and measures. With respect to the quantitative phase that allows one to derive concise information concerning the general data features, the qualitative evaluation better highlights relations among data, giving to the user also the possibility of focusing the attention on specific data subset.

Although in the real experiment we have examined several DW ipercubes, some at different levels of detail also performing slice and dice operations, in the following we provide only an example demonstrating the effectiveness of our technique in highlighting particular data features.

In particular, we compare three ipercubes that are equivalent from a semantic point of view, but characterized by different data distributions since referring to data stored into a different DW (i.e., DW01, DW02 and DW03, see Figure 13). The considered ipercubes are characterized by the attributes Product, Sold Date and Customer as dimensions, and Product Sold as measure. Moreover, the counting function is used for aggregating the data. As a result, in the resulting visualizations
the cyan color is used for identifying parts of the ipercube characterized by a limited number of records where the attribute *Product Sold* does not assume a null value. On the other hand, red cubes identifying parts of the information space characterized by an higher number of records. If all records falling into a specific part of the ipercube are characterized by null values for the selected measure, the corresponding volume in the three dimensional visualization is completely transparent.

Starting from these considerations, different data behaviors outcrops from the visualizations derived by ELDA (in Figure 13 the three columns refer to visualizations concerning different DWs, while the rows display the same representation observed by two different points of view).

In particular, by observing Figure 13(a) one can easily note that DW01 stores data that are not equally distributed into the time domain (Y axis of the representation), since there is a period where any data has been recorded by the information system. From data distribution point of view in the time domain, the other two DWs do not exhibit such behavior.

On the other hand, the representation of the DW02 ipercube highlights a different data distribution feature: from such visualization one can identify an undesirable data behavior (i.e., an evident data clusterization) for the attribute *Product*. By examining the visualization in more detail (i.e., using the detail-on-demand technique), we discovered that most records are characterized by the same attribute value corresponding to the default value that is assigned to the attribute when the user using the information system does not fill the corresponding field. We discovered a similar but less evident behavior in the DW01 ipercube, since also in this case ELDA highlighted an interval (corresponding to the default attribute value) into the *Product* domain where most data are recorded. However, data stored into the remaining part of the domain is more dense with respect to the previous case.

We have also employed constraints viewpoints for better examining data distributions consid-

---

*Figure 13. Visualizing semantically equivalent ipercubes of (a) DW01, (b) DW02 and (c) DW03*
considering different values of one dimension. In the following, we consider only the DW01 ipercube characterized by the attributes **Product Class**, **Sold Date** and **Broker** as dimensions, and **Product Sold** as measure. By observing the ipercube representation using a viewpoint constrained to the upper bound of the attribute **Sold Date** and changing such value, the user can interactively examine data distribution in different time of year, as depicted in Figure 14 (where the color indicates the average number of products sold). Additionally, some preliminary information concerning the data analysis can be derived by comparing Figure 14 (a), (b) and (c). In particular, one can easily derive that:

- There is a broker (corresponding to the last column on the right) that has sold most products, independently from their class;
- There are some product categories that have been sold more than other classes (corresponding to the second, fifth, seventh and ninth rows), independently from a specific broker.

**CONCLUSION**

In this chapter, we have proposed an interactive methodology supporting the DW design and evaluating from both quantitative and qualitative point of view the quality of taken design choices. For the quantitative evaluation we propose a set of metrics measuring different syntactical and statistical data features, while for the qualitative evaluation we propose visualization and interaction techniques effectively supporting data exploration and examination during the DW design phase. Our solution has been successfully experimented on three real DWs, the experimental evaluation demonstrated the effectiveness of our proposal in providing both quantitative and qualitative information concerning the quality of taken design choices. For example, from a quantitative point of view, computed indexes correctly highlighted some inappropriate initial DW design choices. On the other hand, the qualitative evaluation allowed us to interactively examine data distributions more in detail, discover peculiar data behaviors and study relations among selected DW dimensions and measures, as described in Section “”.

Since we have experimented the quantitative phase using unitary values for the metrics coefficients (i.e., 1 or -1), we are currently investigating if an accurate tuning of coefficients allows the procedure to further increase its effectiveness. Moreover, we are investigating if the conditional entropy and mutual information can be used for automatically discovering correlations among...
attributes in order to enable our methodology to suggest alternative design choices during the DW creation. For example, an attribute could represent a valid alternative to another attribute if (i) it is strongly correlated with the second attribute and (ii) its quality is higher with respect to the one measured for the second attribute.

We have recently started at testing our metrics on completely different contexts for evaluating if its effectiveness is independent from the specific application domain; we then shift from ERP systems to applications collecting information on undergraduates, university employers and professors and to DB of motorways crashes. This evaluation is also targeted at highlighting possible limitations of the proposed methodology and can elicit new requirements.

Although designed for providing the user with qualitative information concerning data distributions, we have recently started at evaluating the effectiveness of adopting the proposed three dimensional representation and related interaction techniques not only for the design, but also for data analysis. More specifically, we intend to identify main limitations of our proposal and novel functionalities to be included into ELDA for improving its effectiveness from data analysis point of view.

REFERENCES


