Analyzing Mappings and Properties in Data Warehouse Integration

Domenico Beneventano, Marius Octavian Olaru, Maurizio Vincini[*]

Department of Engineering "E. Ferrari", University of Modena and Reggio Emilia, Italy.

Received 18 July 2016; received in revised form 16 April 2017; accepted 29 April 2017

Abstract

The information inside the Data Warehouse (DW) is used to take strategic decisions inside the organization that is why data quality plays a crucial role in guaranteeing the correctness of the decisions. Data quality also becomes a major issue when integrating information from two or more heterogeneous DWs. In the present paper, we perform extensive analysis of a mapping-based DW integration methodology and of its properties. In particular, we will prove that the proposed methodology guarantees coherency, meanwhile in certain cases it is able to maintain soundness and consistency. Moreover, intra-schema homogeneity is discussed and analysed as a necessary condition for summarizability and for optimization by materializing views of dependent queries.

Keywords: Data Warehouse, information sharing and integration, dimension mappings

1. Introduction

Data Warehouse (DW) is intended as a method for obtaining strategic information from the operational data to be used by business people as an instrument for better understanding the organization's processes. Kimball defines it as a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision-making process [1]. Aside from other properties, we observe that a DW usually integrates data obtained from multiple repositories, which goes through a series of operations (extract-transform-load, or ETL) before being loaded into the final DW. Data is also cleansed to augment its quality, to reflect the decision making process' relevance within the organization. A DW is possibly divided into one or more Data Marts (DM), which are multi-dimensional repositories of analysis of specific aspects of the organization's activities, like services, revenue, shipments, etc., and is usually confined within one single company.

Subsequently, there has been the case where companies needed to combine information obtained from different, heterogeneous DWs to fulfil managerial demands imposed by business decisions or simply by the economical context. For example, it is common nowadays for two or more companies to merge or to collaborate within dynamic business structures (like federation of enterprises, virtual organizations [2], etc.) where managers and business stakeholders need to share and access the common knowledge of the organizations' activity. For this purpose, distributed and independent DW architectures (e.g. [3]) and networks (e.g. [4]) have been proposed to allow common interest information sharing and integration among distinct stakeholders.

In this paper, a mapping-based integration methodology that is able to generate semantic mappings between dimensions of different DWs is proposed, either between dimension categories or between members of such categories. The work is motivated by data-quality which is a crucial aspect when building an inter-organization DW; in fact, given the use of the DW, incorrect or low-quality data may not only make the DW useless, but its use may also lead to wrong decisions with potential negative impact on the organization. For these reasons data-quality requirements for the generated mappings, such as coherency and consistency, are formally analyzed to ensure that the integrated information can be correctly aggregated (or

[*] Corresponding author. E-mail address: Domenico.Beneventano@unimore.it

disaggregated). This observation is relevant as the multidimensional data is usually explored along aggregation patterns, drilling-down or rolling-up from a starting analysis point. With respect to the previous works [5,6] where the mapping-based integration methodology was introduced, the present paper expands such methodology and performs extensive property analysis of the mapping discovery and integration techniques.

The paper is structured as follows: Section 2 provides an overview of related work. Section 3 provides the preliminary discussion of dimension mappings, the properties that a mapping has to guarantee, and also a discussion on intra- and inter-schema heterogeneity; Section 4 describes the mapping and integration methodology to be analyzed, while Section 5 provides the analytic discussion of the properties that are guaranteed and/or maintained while performing mapping discovery and dimension integration. Finally, Section 6 contains the conclusions of the current work.

2. Related Work

As stated in [7], the problem of data warehouse integration has received less attention than the general problem of databases integration, which has been extensively studied in the literature ([8-10]). As a consequence, few approaches that deal with the data warehouse integration problem systematically, and fewer still that attempt to provide a complete integration methodology, are available.

To tackle the basic issue of matching heterogeneous dimensions, the work in [11] introduces the desirable properties of coherence, soundness and consistency that "good" matchings between dimensions should enjoy and, then, proposes two different approaches to the problem of integration that try to enforce matchings satisfying these properties; the authors have also shown that such properties give the possibility to correlate, in a correct way, multiple data marts by means of drill-across queries, which are basically joins, over common dimensions, of different data marts. In this present paper an integration methodology is proposed; the metodology is based on the properties of coherence, soundness and consistency, which is substantially different from the one proposed in [11]. First, unlike [11] where matchings are provided, i.e., matchings are an input of the problem; in the proposed methodology matchings that identify similar categories from the two dimensions are generated. Moreover, and more important, in the proposed methodology each dimension is augmented with compatible categories and members from the other dimensions and a formal discussion of how properties are preserved along this technique is provided. The preliminary idea to map and import categories and members of different dimensions to allow information integration between two or more compatible dimensions was introduced in previous works [5, 6].

In [12] the authors define the term "conformed dimensions" as either identical or strict mathematical subsets of the most granular and detailed dimensions. Conformed dimensions share dimension keys, column names, attribute definitions and attribute values. Conformed dimensions are, of course, coherent, and in some cases sound and consistent. However, rather than providing an integration methodology, the author defines the "Data Warehouse Bus Architecture", which is a design methodology for incrementally building the enterprise Data Warehouse to facilitate the integration of autonomous Data Marts sharing the conformed dimensions.

In [13], the authors provide a mapping technique for DW elements based on semantics and on earlier work in data integration ([8, 10, 14]); class similarity is used to find related elements (facts, dimension, and aggregation levels) that the authors use to generate mappings between two DW schemas; the mapping derived for dimension categories is conceptually equivalent to the mapping proposed in [11]. The authors also discard mappings that are not coherent and study the stable marriage problem for selecting the best mapping from more than one candidate. The method proposed in this paper is different from the semantics-based method presented in [13] because it uses structural and cardinality-based properties to generate dimensional mapping, while semantics may be used as a validation step. Moreover, rather than simply discarding non-coherent mappings, the proposed methodology directly generates mappings that are coherent.

Properties of dimensions have also been defined as normal forms for multidimensional databases [15]. Normal forms are necessary to ensure good design qualities for multidimensional data, guarantee summarizability and lack of redundancies in the multidimensional database. Without stretching the analysis deeper, in the present paper the dimensional normal form [15] will be analysed when integrating heterogeneous dimensions.

3. Preliminaries

Many models and approaches have been proposed for the conceptual design of a DW and for deriving multidimensional schemata from E/R or relational schemata, XML, ontologies, web and other structured/semi-structured or unstructured data stores, although most of them share the same basic ideas of data (facts) organized along dimensions of analysis, that are usually hierarchies of aggregation levels used to interpret the information at various aggregated views. The facts are analyzed using numerical measures that express a property of interest. This multidimensional view of data fits well the needs of analysts and designers, and it constitutes an intuitive method for graphically representing concepts of interest both for developers and for business people. The dimensions are represented as hierarchies of aggregation levels, usually called categories or dimensional attributes that are populated by members or values. For example, in a temporal dimension like the one proposed in Fig. 1, year is a category, while {2010; 2011; 2012} are members of the category year.

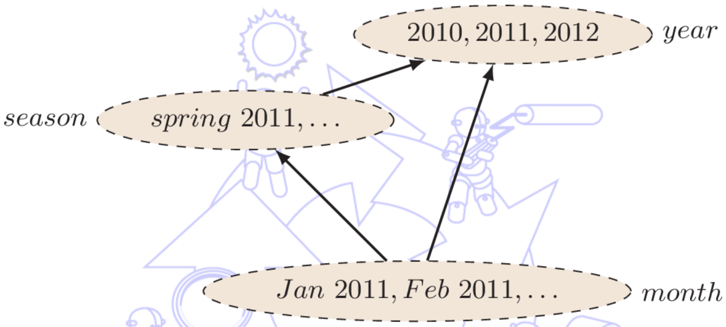

Fig. 1 A *time* dimension

For the purpose of this paper, the formalization proposed in [16] is used, as it allows us to formally reason both on dimension schemas and instances.

A dimension schema is a directed a cyclical graph (dag) H = (C; $\nearrow$), where C is a finite set of categories, having a distinguished category All $\in$ C that is a sink (it is a vertex of the graph that is reachable from every other node through an arc or a path). The partial order relation $\nearrow$ expresses the conceptual roll-up relations among categories. A bottom category is a category c bottom that is reachable from no other node of the dag through an arc or a path (i.e., there is no c $\in$ C such that c $\nearrow$ c bottom). For the purpose of this paper, a single bottom category for dimension will be assumed. Note that various models that allow a dimension to have more than one bottom category (e.g. [17]) have been proposed in literature; however we believe that dimensions with only one bottom category allow a cleaner representation of multidimensional data.

An example of dimension schema is presented in Fig. 1, where [month $\nearrow$ season] and [month $\nearrow$ year] (for brevity, the category All has been omitted).

A hierarchy domain is a dag h = (M; <), where M is the set of members of the hierarchy, with a distinguished member all $\in$ M that is a sink. The members are also organized in a graph structure (< is a partial order relation on the set M) to express the roll-up relations among the categories they belong to.

A dimension d over a schema (C; $\nearrow$) is a graph morphism d: (M;<) → (C; $\nearrow$) such that: (a) d(all) = All, and (b) $\forall$x; y; z such that x <* y, x <* z (<* is the transitive and reflexive closure of <) and y $\neq$ z, then d(y) $\neq$ d(z). Let m: C → P (M) be a function that assigns each category the set of members of that category; in other words m(c) = {m $\in$ M | d(m) = c}, for every c

∈ C. Occasionally, m(c) will be called "members of c". The hierarchy domain may also be described by a family of roll-up functions [18] ρ^(c_1 →⟦ c⟧_2) : m(c_1) → m(c_2); the roll-up functions are defined for all c_1 ↗ c_2 as ρ^(c_1 →⟦ c⟧_2) (m1) = m2 for all m1 ∈ m(c_1) and m_2 ∈ m(c_2) such that $m_1 < m_2$.

3.1. *Dimension Mappings*

A dimension mapping (in its simplest form) is a function that maps categories of one dimension to categories in the other dimension [19]; in other words, given two dimensions d1: (M_1; <1) → (C_1; ↗1) and d2: (M_2; < 2) → (C_2; ↗ 2), a mapping is a function μ: $C_1 \rightarrow C_2$ that may be total or partial. The mapping identifies for some/all categories in d1 a category in d2 that expresses the same concept at the same level of detail. Like in traditional data integration, the mappings may be used to express semantic similarities among different structures, and integration-wise may be used to rewrite queries over compatible schemas and instance [20] and to integrate the information obtained from different instances.

3.2. *Dimension Mapping Properties*

The works in [11, 18] define three properties that matching among dimensions may have: coherency, soundness and consistency.

Coherency is a property concerning the dimension schemas, in particular the partial order relation imposed on the category set. A matching μ: $C_1 \rightarrow C_2$ is coherent if ci ↗1* c_j ⇔ μ (c_i) ↗ 2* μ (c_j), where ↗ 1* and ↗ 2* are the reflexive and transitive closures of ↗ 1 and ↗ 2, respectively. The coherency property states that the roll-up relations among attributes are maintained through the mapping between the dimensions.

Soundness is a property concerning the members of categories, in particular a matching is sound if m (c) = m (μ(c)) for all c ∈ C_1. The property guarantees that mapped categories contain the same members.

Consistency ensures that the roll-up function between members is maintained, that is ∀ mi1, mj1 ∈ M1 such that m_{i1} <1* m_{j1} then ∃ m_{i2} ∈ m(μ(d(m_{i1}))) and m_{j2} ∈ m(μ(d(m_{j1}))) such that m_{i2} <2* m_{i2}, where <1* and <2* are the transitive closures of <1 and <2. Alternatively, ρ^ (c_1 →⟦ c⟧_2) = ρ^ (μ (c_1) →⟦μ (c⟧_2)) for every c_1, c_2 ∈ C1.

In other words, if two members roll-up in one dimension, then the mapping should guarantee that there are two equivalent members in the other dimension that maintain the roll-up relation. Furthermore, a mapping that is coherent, sound and consistent is called a perfect matching.

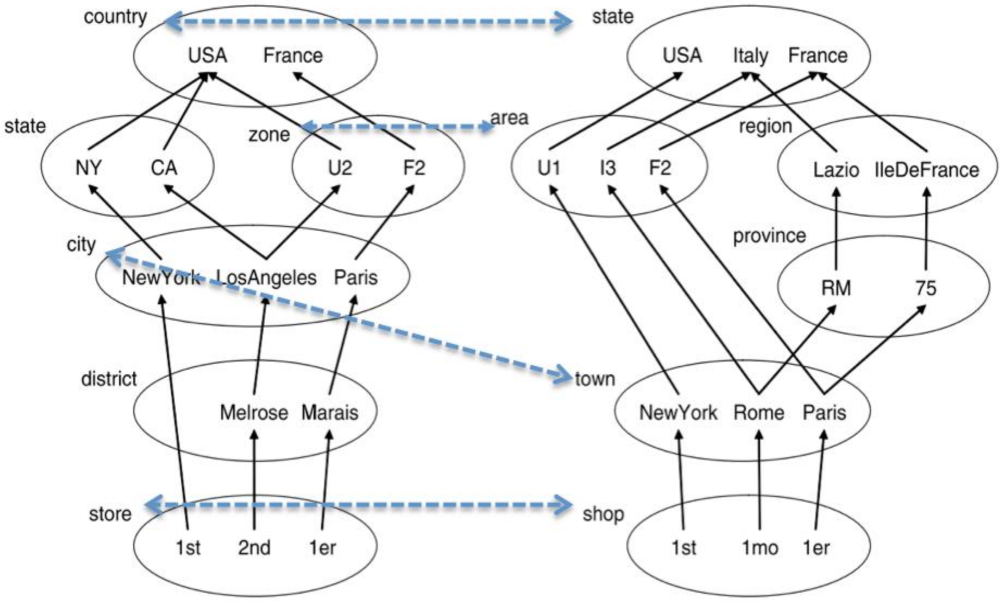

Fig. 2 A matching between two dimensions (dashed arrow) and possible instances of the dimensions

In Fig. 2, the example introduced in [11] is considered; the matching between the two dimensions d_1 (on the left) and d_2 (on the right), represented by dashed arrow, is the following:

μ(country) = state

μ(zone) = area

μ(city) = town

μ(store) = shop

This mapping is clearly coherent since, roughly speaking, there are no “cross mappings” i.e. no correspondence intersects another. The matching is consistent, since the roll-up functions are not contradictory, but it is not sound, since there are members in one dimension that do not appear in the other.

3.3. *Homogeneous and Heterogeneous Dimensions*

Studying the DW integration problem, it is interesting to analyze the different kinds of heterogeneities that sources may contain and the way they are eliminated/introduced by applying the integration methodology.

Heterogeneity is generally used to define systems that are in some way different, with distinct data/information or different ways of representing the same information. As there is no general consensus as how to classify homogeneity/heterogeneity, we distinguish two types that may apply in DW integration: intra- and inter-schema heterogeneity.

Intra-schema heterogeneity is used to define roll-up inconsistencies inside a single schema. In [16] a schema is defined homogeneous if and only if all the members in a category c_1 roll up to a member of every category c_2 such that $c_1 \nearrow c_2$. Furthermore, if a schema is homogeneous and has only one bottom category (like the ones assumed throughout the current paper), then it is called strictly homogeneous. If a schema is not homogeneous, then it is called heterogeneous. This kind of homogeneity will be referred as intra-schema homogeneity (the opposite being intra-schema heterogeneity). Initial multidimensional models were homogeneous, but the restriction has been later dropped to allow a more compact dimension design and a more efficient space allocation due to the lower number of categories [16]. On the other hand, homogeneity allows a clear representation of a hierarchy schema as it provides an intuitive representation of the aggregation levels inside one single dimension. In fact, for every two categories c_1 and $c_2 \in C$, a roll-up relation $c_1 \nearrow c_2$ clearly states the completeness of the aggregation function, as every member of c_1 is related (or aggregated) to a member of c_2.

Intra-schema homogeneity is also important when analyzing dependent GROUP BY queries. In [21], two queries Q1 and Q2 are dependent if the results of one can be computed from the results of the other. For example, when analyzing the sales of one company, if every sale is related to a city and every city belongs to a region, then the total revenue for the sales grouped by region (call it query Q1) may be obtained from the revenue grouped by city (query Q2) by adding the revenue for every city inside one single region. This observation is important when performing optimization on the DW by materializing frequently accessed views [21] or by pre-computing aggregations when using the CUBE operator.

Intra-schema homogeneity is also a necessary condition for summarizability, which has been defined in the field of statistical databases as the ability to correctly compute data aggregated at a certain category (called classification node, or C-node) from the same data aggregated at a lower category [22]. Furthermore, by using the closed-world assumption (part of the completeness hypothesis), a homogeneous dimension with only one bottom category is in dimensional normal form (DNF), as defined in [15]. Checking homogeneity can thus assert important quality properties for the final schema and instance after applying the integration methodology.

In this paper, intra-schema heterogeneity is formally analyzed and a discussion about how it is maintained when integrating two or more DW dimensions is presented. In general, homogeneity and heterogeneity are not preserved when integrating two distinct DWs.

With the integration methodology proposed in this paper it is possible to obtain a heterogeneous dimension from two homogeneous dimensions, or a homogeneous dimension from two heterogeneous dimensions (see Section 5.3).

Inter-schema heterogeneity is a concept derived from database theory and used to describe two or more different data sources that are in some way distinct by schema or by form (see [3] for a discussion about heterogeneities).

Following the same approach, two or more different DWs may contain the same or similar information, but differently structured or identified by different instance values. For example, hierarchies may contain similar information but structured differently (higher granularity, inconsistent roll-up functions, different category names); the same information may be represented as a fact attribute in one schema or as a category in another; or the schemas may contain different and possibly inconsistent measures that are incompatible/inapplicable on all the DWs. Some researchers (for example, [13]) have attempted to classify the different kinds of heterogeneity that may occur between two DW instances.

Finally, we point out that some approaches are used to eliminate inter-schema heterogeneities by integrating or unifying two or more different DW dimensions into one single dimension embedding the two initial heterogeneous dimensions.

In the approach proposed in this paper, inter-schema heterogeneity is implicitly reduced or eliminated by performing dimension integration. One particular case discussed in Section 5.3 allows the complete elimination of inter-schema heterogeneity, by rendering two dimensions identical after performing the integration methodology.

4. Integrating Heterogeneous Dimensions

Let's assume a scenario where multidimensional data obtained from two or more DWs sharing common/similar dimensions must be integrated. The current section describes how the integration methodology introduced in [5, 6] can be used to map and import categories and members of different dimensions to allow information integration between two or more compatible DW dimensions. The methodology consists of the following steps:

- **Mapping categories**: a mapping ξ that identifies similar categories from the two dimensions is generated;
- **Importing categories and members**: each dimension is augmented with compatible categories and members from the other dimensions; new mapping $\xi^{\#}$ is derived from ξ.

4.1. Mapping Categories

To map similar categories of different dimensions, standard approaches like semantics may be used [23]; however in accordance with other researchers [8] we believe that a systematic approach that considers all structures of the DWs may yield better results. In fact, despite numerous proposals, almost all design methodologies consider the facts of interest analyzed along dimensions composed of different aggregation levels (or categories), in graph-like structures. If the instance contains complete information, the dimensions itself embed sufficient information to allow the automatic or semi-automatic mappings discovery.

The main observation is that similar or identical information is usually structured in similar ways even by different working groups, in accordance with the common view of the information of interest. The simplest examples are time and space dimensions. A time dimension will surely contain days grouped into months that are grouped into trimesters, semesters, years and sometimes even into decades; this aggregation hierarchy must be identical even in different DWs because this division reflects the way time is universally organized and understood. Similarly, addresses are associated to a city; cities are organized into communes, grouped into regions, countries, and so on.

The concept may also be observed when representing other kind of information. Consider, as an example, two companies managing health data. The various health conditions will certainly be categorized in a similar manner, in accordance with the common representation of the knowledge of interest (for example, the World Health Organization provides an International Classification of Diseases - ICD). When integrating different data marts within the same organization there will likely be common dimensions that reflect the way the company is organized.

For example, employees are organized in groups that belong to departments that have a manager, and so on. To map similar categories, we consider the directed graph representing the dimension schema and a property called cardinality-ratio, which is the ratio of the total numbers of distinct members of two related categories. Formally, given a dimension d: (M; <) → (C; ↗), for every two categories ci and c_j ∈ C, the cardinality-ratio $\tau_{c_i \to c_j}$ among the two categories is defined as:

$$\tau_{c_i \to c_j} = \begin{cases} \frac{\#m(c_j)}{\#m(c_i)} & if\ c_i \nearrow^{*} c_j\ and\ c_j \neq c_j \\ 1 & if\ c_i = c_j \\ 0 & elsewhere. \end{cases}$$

For example, in a time dimension, assuming there are two categories, month that rolls-up to year, then $\tau_{month \to year} = 12$, as a member of the category year is an aggregation of 12 members of the category month, assuming the instance contains all the months of every year.

Consider two dimensions that must be mapped: d_1: (M_1; <1) → (C_1; ↗ 1) and d2: (M_2; < 2) → (C_2; ↗ 2). The proposed mapping generating methodology considers the two dimension schemas and annotates each label with its cardinality ratio. Two new labeled graphs are derived, G_1 = [(C_1, ↗ 1*); f1] and G2 = [(C_2, ↗ 2*); f_2], where ↗1* is the transitive and reflexive closure of ↗ 1 (similarly ↗ 2*). The function f_1 is defined as follows (the function f_2 is similarly defined):

$$f_{1\,:} \nearrow_1 * \longrightarrow \mathbb{Q}^{+}$$

$$f_1\,[c_i, c_j] = \tau_{c_i \to c_j}$$

The function f_1 is transitive, meaning that f1 [c_i, c_k] = f_1 [c_i, c_j] x f_1 [c_j, c_k], for all c_i,c_j ,c_k ϵ c_1 such that c_i ↗ 1* c_k and c_k ↗ 1* c_j. For example, in Fig. 3, f_1 [i_1, i_3] = f_1 [i_1, i_2] x f_1 [i_2, i_3] (green arrow). This ensures that the cardinality-ratio applied to the transitive and reflexive closure of the roll-up relation ↗ 1 is correct. Graph theory is then used to derive a maximum rank graph G that is sub-graph isomorphic to both G_1 and G_2.

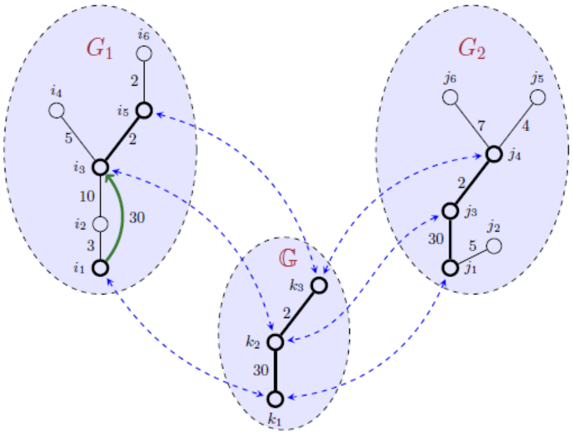

Fig. 3 Mapping generation

The maximum rank subgraph is a graph with the highest number of nodes. Note that the case where more than one maximum rank subgraphs may be derived is not considered. In such situations, analysts may decide to use independently one, more than one, or even all of the subgraphs, if the generated mappings do not conflict with each other. A graph T is sub-graph isomorphic to a graph U if there exists a subgraph of U that is isomorphic to T.

Based on the subgraph G, a mapping function ξ: $C_1 \rightarrow C_2$, which may be total or partial, is derived to associate categories of the initial dimension schemas through the subgraph isomorphism (see Fig. 3, where for the sake of simplicity, the Figure does not depict the complete transitive and reflexive closure of the schemas). Of course, there may be the case where the dimensions are completely distinct or the instances are incomplete (for example, not all the months of every year are contained in the instance, so the cardinality-ratio is different among pairs of similar categories in the two dimensions) in such way to compromise the mapping generating step that is either incapable of generating mappings or generates incorrect ones. For this latter case, semantic validation to increase the accuracy of the mappings by discarding possibly incorrect ones is added in [5].

4.2. Importing categories and members

The mappings generated in the previous step are useful for integrating information coming from the various instances and in some cases to write queries and to reformulate them over the different instances. This capability is, however, limited by the differences over the DW dimensions. To overcome this shortcoming, the methodology presented here includes a category importation step intuitively depicted in Fig. 4 and defined as follows.

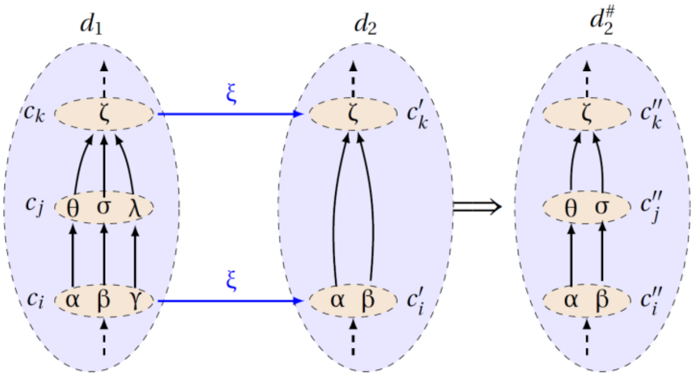

Fig. 4 The *category* and *member* importation rule

Let d_1 and d_2, be two dimensions, and c_i, $c_j \in C_1$ such that $c_i \nearrow 1$ c_j and $c_k \in C_2$. If $\xi(c_i) = c_k$ and ξ(cj) $\notin$ C_2, then d_2 is augmented with the category c_j and with the roll-up relations derived from the semantic mappings. Thus, a new dimension $d_2^\#$: $(M_2^\#, <_2^\#) \rightarrow (C_2^\#, \nearrow_2^\#)$ is derived, where $C_2^\# = C_2 \cup \{c'_j\}$, $\nearrow_2^\#$ is extended to include the relations between c_j and the categories of C_2, and $M = M_2 \cup \{m_{c_j}\}$. The mapping ξ is extended to include the newly imported category. A new mapping $\xi^\# : C_1 \rightarrow C_2^\#$ is generated as follows:

$$\xi^\#(c) = \begin{cases} \xi(c), & if\ c \neq c_j \\ c_j, & if\ c = c_j \end{cases}$$

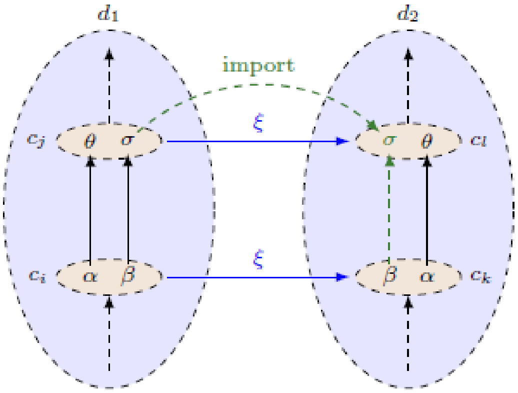

Fig. 5 No sound → sound mapping

The members of c'_j are imported using an approach based on the RELEVANT [24] clustering methodology and the relation <2 is extended (call $<_2^\#$ its extension) to include the relations between the newly imported members and the initial members of the categories in d_2. Formally, for every c_i, $c_j \in C_1$ and c_k, $c_j' \in C_2^{\#}$ such that $c_i \nearrow 1\ c_j$ and c_k, $\nearrow_2^{\#}\ c_j'$, for all $m_i \in m(c_i)$ and $m_j \in m(c_j)$ such that $m_i <1\ m_j$, and for all $m_k \in m(c_k)$ such that $m_i = m_k$, then it must be that $m_j \in M_2^{\#}$ and $m_k <_2^{\#}\ m_j$ (see Fig. 4). In other words, the newly imported category c_j' is populated with some/all of the members of c_j and also the roll-up relations between the members of c_i and c_j are inherited into d_2.

This observation will be later used when proving the preservation of soundness and consistency of mappings generated by the presented methodology. Note that the member import rule may be used in a broader sense for categories already contained in the dimensions d_1 and d_2. By replacing the imported category c_j' with any other category $c_l \in C_2$ such that $c_k \nearrow 2$ cl and ξ (c_j) = c_l, then the member import rule gives the capability of augmenting the domain of category c_l, thus increasing the information contained in dimension d_2. This particular case will be used to reason about the properties of the dimensions in Fig. 5.

For the sake of simplicity, the above example does not highlight the advantages of using RELEVANT, which was chosen for its ability to combine information from more than one dimension and to discriminate between incorrect members by using clustering techniques rather than direct equality of the members.

As an example, let us consider the two dimensions and the matching of Fig. 2. If we apply the member importation rule shown in Fig. 4, the category district of the dimension d_1 is imported into the dimension d_2: in this way, users of the DW with the dimension d_2 can now compute aggregated information grouped by district, for some shops (the ones also present in the dimension d_1). The result is shown in Fig. 6, where the hierarchy for dimension d_2 is drawn; the figure also contains an example of relationships among members: in parentheses, the members related to the shop 1er are shown.

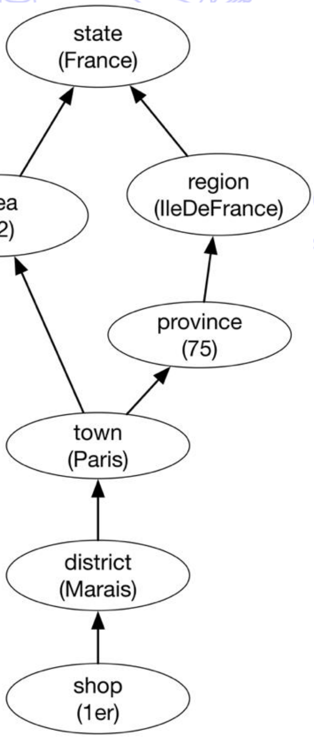

Fig. 6 Example of importation rule application

5. Property Analysis

This section will analyze the properties that are guaranteed and preserved when performing mapping discovery and category and member importation using the methodology in Section 4. Most notably, the mapping generation step guarantees coherency, meanwhile the importation step preserves soundness and consistency and in some cases may render a mapping that is neither sound nor consistent into a mapping that is sound and/or consistent. In the remainder of the section, let us assume that ξ is a mapping” generated by the first step of the methodology presented in Section 4.

5.1. *Coherency Check*

One important result of Section 4 is that coherency is unconditionally guaranteed.

Theorem 1. *The mapping ξ is coherent.*

The theorem may be proved by using the graph isomorphisms that in a directed graph preserve node links (and paths) and their order.

Proof. Let G = [(C, ↗); f] be the graph that is subgraph-isomorphic to both G_1 and G_2. Then C_{T4}∃C_2⊆∃ $C_{T_1} \subseteq C_2$ such that $G_{1\,|C\,T_1} = [(\nearrow_{1\,|\,T_1}), f_{1\,|\,C\,T_1}]$ and $G_{2\,|C\,T_2} = [(\nearrow_{2\,|\,T_2}), f_{2\,|\,C\,T_2}]$ are isomorphic to G, where $G_{1\,|C\,T_1}$ and $G_{2\,|C\,T_2}$ are the restrictions of G_1 and G_2 to CT_1 and CT_2 respectively. It follows that there are two graph isomorphisms w_1: $G_{1\,|C\,T_1}$ → G and w_2 : $G_{2\,|C\,T_2}$ → G. Let $w = w_2^{-1} \circ w_1$; w is also a graph isomorphism from $G_{1\,|C\,T_1}$ to $G_{2\,|C\,T_2}$.

The graph isomorphism ensures that for all c_i and $c_j \in CT_1$ it stands that $c_i \nearrow 1^* c_j \Rightarrow$ w $c_i) \nearrow 2^*$ w(c_j). Let $\xi_{|\,C_{T_1}}$ be the restriction of ξ to CT_1. By construction, $\xi_{|\,C_{T_1}} \equiv w$. It follows that $c_i \nearrow 1^* c_j \Longrightarrow \xi_{|\,C_{T_1}}(c_i) \nearrow 2^* \xi_{|\,C_{T_1}}(c_j)$.

Thus, $\xi_{|\,C_{T_1}}$ is coherent. By extension, ξ is also coherent.

5.2. *Soundness and Consistency Check*

Although the first step of the integration methodology produces a coherent mapping, soundness and consistency are guaranteed only in certain cases. To verify whether soundness is verified, two steps must be performed. First, the initial mapping must be checked. Formally, for all categories c ∈ C_1, it must be that m(c) = m ξ(c)). This is a simple inclusion test that will be analyzed no further. Secondly, the soundness and consistency property must be verified when performing the category and member importation.

The following theorem provides a sufficient condition for guaranteeing soundness and consistency when importing categories and members.

Theorem 2. If *ξ is sound and consistent*, then $\xi^{\#}$ is also *sound and consistent*.

Proof. If ξ is sound, then m(c_i) = m(c_k). The member importation rule states that if $c_i \nearrow 1^* c_j$ and $c_k, \nearrow_2^{\#} c_j'$ and $\xi^{\#}$ (c_i) = c_k, then for all $m_i \in$ m(c_i), $m_j \in$ m(c_j) and $m_k \in$ m(c_k) such that mi <1 m_j and $m_i = m_k$, then it must be that: (a) $m_j \in$ m(c_j'); and (b) $m_k <_2^{\#} m_j$ (see Fig. 3).

If ξ is sound, from (a) follows that $\xi^{\#}$ is also sound.

If ξ is consistent, from (b) follows that $\xi^{\#}$ is also consistent.

Theorem 2 provides a sufficient, but not necessary condition for soundness and consistency. In fact, there may be cases when mapping two dimensions where the initial mapping ξ is neither sound nor consistent, but the final mapping $\xi^{\#}$ becomes sound and/or consistent after the second step of the integration methodology. For example, Fig. 5 provides two dimensions and a mapping ξ that is neither sound nor consistent, as m(c_j) ≠ m(c_l) (the member β belongs to m(c_j) but not to m(c_l)) and $\rho^{c_i \to c_j}$ ≠ $\rho^{c_k \to c_l}$ (member rolls-up to member in dimension d_1, but rolls-up to no member of cl in dimension d_2). Note that dimension d_2 is heterogeneous. Assuming the methodology generated the mapping ξ (see Fig. 5), step 2 of the methodology renders the mapping sound and consistent.

The reason soundness and consistency are considered together is that they are closely related. In some cases (not all) soundness follows from consistency.

The following corollary states a relationship between consistency and soundness of a mapping relation.

Corollary 1. *If ξ maps only pairs of categories c_i and c_j such that $c_i \nearrow c_j$ and ξ is consistent, than ξ is also sound.*

Proof. Let c_i, $c_j \in C_1$ such that $c_i \nearrow 1\, c_j$. By consistency, it must be that $\rho^{c_i \to c_j} \equiv \rho^{\xi_{c_i} \to \xi_{c_j}}$, that requires that Dom($\rho^{c_i \to c_j}$) = Dom($\rho^{\xi_{c_i} \to \xi_{c_j}}$) and Codom($\rho^{c_i \to c_j}$) = Codom($\rho^{\xi_{c_i} \to \xi_{c_j}}$). The conditions are equivalent to m (c_i) = m (ξ(c_i)) and m (c_j) = m (ξ(c_j)); thus soundness is proved.

Corollary 2. *If ξ is a perfect matching, then $\xi^{\#}$ is also a perfect matching.*

Proof. The proof follows from Theorems 1 and 2.

5.3. Checking Homogeneity

Unfortunately, homogeneity is not preserved when integrating different DW dimensions. Not even the case when integrating two homogeneous dimensions can ensure that the derived dimension is homogeneous. For example, in Fig. 5 the mapping ξ establishes semantic equivalences among categories belonging to the homogeneous dimensions d_1 and d_2. When importing members from dimension d_1 to dimension d_2, the newly derived dimension, $d_2^{\#}$ is heterogeneous (the member γ has no equivalent member m_j in the category c_j'' such that $\gamma <2\, m_j$ and rolls-up directly to a member in category c_k'').

Interestingly, heterogeneity is also not preserved. There may be the case when a homogeneous dimension is obtained when integrating two heterogeneous dimensions. For example, in Fig. 7 when integrating members from dimension d_1 to dimension d_2 (both heterogeneous), the newly derived dimension $d_2^{\#}$ is homogeneous. In this later case, the instance of dimension d_2 is completed with information from d_1. A situation like the one described in Fig. 7 may be encountered in real life cases when analysts decide to model the same information differently, or when information is partially missing by choice or by error. For example, categories c_j and c_j' may represent the region of a city (categories c_i and c_i'), that was omitted for some cities in d_1 (member) or for other members in d_2 (member). The missing information is thus derived from the other dimension.

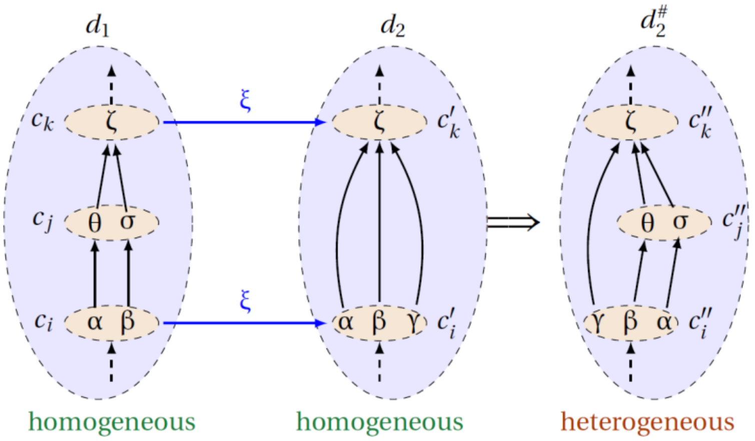

Fig. 7 Homogeneous → Heterogeneous

In some circumstances, homogeneity may be maintained when integrating two different homogeneous dimensions. The following theorem provides a sufficient condition to guarantee the preservation of homogeneity.

Theorem 3. If d_1 and d_2 are homogeneous and m (c_k) $\subseteq$ m (c_i), then $d_2^{\#}$ is also homogeneous.

Proof. Consider the importation rule depicted in Fig. 4. From m (c_k) ⊆ m (c_i) it follows that for all $m_k \in$ m (c_k), there is a member $m_i \in$ m (c_i) such that $m_i = m_k$. Since d_1 is homogeneous, there is also $m_j \in$ m (c_j) such that $m_i <1\, m_j$. By construction, the category and member importation steps build a new dimension $d_2^{\#}$ such that $c_j \in C_2$ (c_j in d_2 will be named c_j', to avoid confusion), $c_k \nearrow_2^{\#} c_j'$ and $m_j \in$ m(c_j') such that $m_k <_2^{\#} m_j$.

Thus is proved that in the dimension $d_2^{\#}$, for all $m_k \in m(c_k)$ there is a member $m_j \in m(c'_j)$ such that $m_k <_2^{\#} m_j$. Thus homogeneity is preserved.

Corollary 3. If d_1 and d_2 are in dimensional normal form and m $(c_k) \subseteq$ m (c_i), then $d_2^{\#}$ is also in dimensional normal form.

Proof. Assuming to be working under the closed-world assumption and assuming only one bottom category per dimension, the proof follows from Theorem 3.

An interesting observation may be drawn from Theorem 3. It turns out that when integrating two homogeneous dimensions d_1 and d_2 with bottom categories $cbottom_1$ and $cbottom_2$, if $m(cbottom_1) = m(cbottom_2)$, then by importing categories and members from one dimension to another, the newly obtained dimensions $d_1^{\#}$ and $d_2^{\#}$ are identical, a part from the names of the categories. In other words, there will be a total matching $: C_1^{\#} \rightarrow C_2^{\#}$ that is perfect. Furthermore, χ -1 is also a perfect matching. The newly derived dimensions $d_1^{\#}$ and $d_2^{\#}$ are identical to the one derived in [11] using the tightly coupled approach.

Corollary 4. If m $(m_k) \nsubseteq$ m (c_i) and $c_j \notin C_2$, then $d_2^{\#}$ is heterogeneous.

Proof. If m $(m_k) \nsubseteq$ m (c_i), then there is $m_k \in$ m (c_k) such that $m_k = \notin$ m (c_i). Implicitly, there is no $m_j \in$ m (c_j) such that m_k <1 m_j. It follows by construction that $\nexists$ $m_j \in$ m (c'_j) such that $m_j <_2^{\#} m_k$. Thus, $d_2^{\#}$ is heterogeneous.

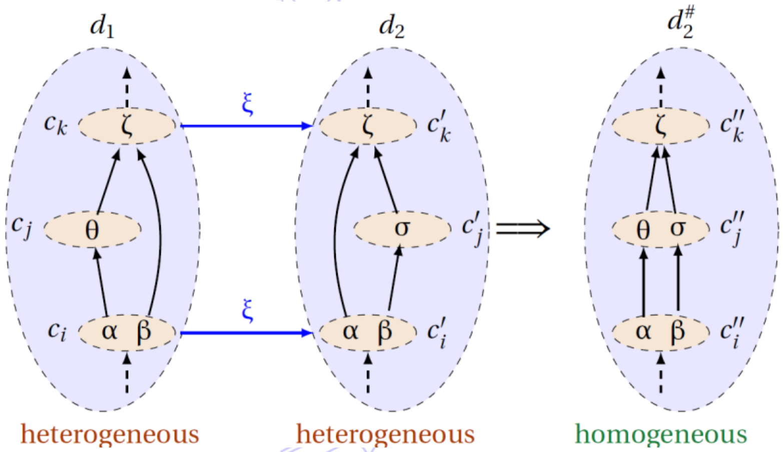

Fig. 8 Heterogeneous → Homogeneous

6. Conclusions

Data Warehouse integration may be performed by using classical data integration-like approaches, by means of semantic mappings that are used to express similarities between different DW elements (dimension categories, facts, etc.) and to combine information from distinct DWs either by directly integrating information from different repositories or by executing drill-across queries. Even if in data-integration partial or not completely accurate results may be used, that is not the case when integrating information from two or more DWs. Given the high quality requirements when building and querying a DW, any mapping-based integration methodology must ensure the correctness and accuracy of the information it integrates.

In the current paper, a formal analysis of a mapping based DW integration methodology and its properties have been performed. The methodology presented here is able to generate mappings that are coherent, which in turn allow correct aggregation of information from the different DWs. Moreover, under specific constraints, after performing the integration steps, the mapping may also be rendered sound and consistent. Coherency and consistency are properties related to roll-up relations, thus a mapping satisfying both properties ensures that the integrated information can be correctly aggregated (or disaggregated). This observation is relevant as the multidimensional data is usually explored along aggregation patterns, drilling-down or rolling-up from a starting point analysis.

On the other hand, soundness states that the mapped dimension categories contain the same members, which in turn give analysts the possibility of executing meaningful drill-across queries that would otherwise be impossible if the related categories contained distinct members.

Finally, although some researchers allow the design of intra-heterogeneous dimensions, on the other hand, we consider that homogeneity allows a clearer representation of multidimensional data, both for designers and analysts as for business people that may have a simpler perception of the underlying DW model. The present paper analyzed intra-schema heterogeneity directly and provided a sufficient condition for maintaining homogeneity that is a necessary condition for summarizability and for materializing views as a mean of optimizing response time when executing dependent queries.

References

[1] W. H. Inmon, Building the data warehouse, 3rd ed. John Wiley & Sons, Inc., 2002.

[2] M. Preis and J. Seitz, "Challenges and conflicts integrating heterogeneous data warehouses in virtual organisations," International Journal of Networking and Virtual Organisations, vol. 11, no. 3/4, pp. 329-335, 2012.

[3] A. P. Sheth and J. A. Larson, "Federated database systems for managing distributed, heterogeneous, and autonomous databases," ACM Computing Surveys, vol. 22, no. 3, pp. 183-236, September 1990.

[4] S. Abiteboul, I. Manolescu, and N. Preda, "Constructing and querying peer-to-peer warehouses of XML resources," Proc. 21st International Conference on Data Engineering (ICDE 2005), IEEE Press, April 2005, pp. 1122-1123.

[5] S. Bergamaschi, M. O. Olaru, S. Sorrentino, and M. Vincini, "Dimension matching in peer-to-peer data warehousing," Proc. IFIP Working Group 8.3 International Conference on Decision Support Systems, 2012, pp. 149-160.

[6] F. Guerra, M. O. Olaru, and M. Vincini, "Mapping and integration of dimensional attributes using clustering techniques," E-Commerce and Web Technologies, Springer press, 2012, pp. 38-49.

[7] R. Torlone, Interoperability in data warehouses, Encyclopedia of Database Systems, Springer, pp. 1560-1564, 2009.

[8] D. Beneventano, S. Bergamaschi, G. Gelati, F. Guerra, and M. Vincini, "MIKS: an agent framework supporting information access and integration," Intelligent Information Agents, vol. 2586, pp. 22-49, 2003.

[9] S. Bergamaschi, G. Gelati, F. Guerra, and M. Vincini, "An intelligent data integration approach for collaborative project management in virtual enterprises," World Wide Web, vol. 9, no. 1, pp. 35-61, March 2006.

[10] A. Halevy, A. Rajaraman, and J. Ordille, "Data integration: the teenage years," Proc. of the 32nd international conference on Very large data bases, September 2006, pp. 9-16.

[11] R. Torlone, "Two approaches to the integration of heterogeneous data warehouses," Distributed and Parallel Databases, vol. 23, no. 1, pp. 69-97, February 2008.

[12] R. Kimball and M. Ross, The data warehouse toolkit: the complete guide to dimensional modeling, New York: John Wiley & Sons, Inc., 2002.

[13] M. Banek, B. Vrdoljak, A. M. Tjoa, and Z. Skocir, "Automated integration of heterogeneous data warehouse schemas," International Journal of Data Warehousing & Mining, vol. 4, no. 4, pp. 1-21 October-December 2008.

[14] D. Beneventano, S. Bergamaschi, F. Guerra, and M. Vincini, "The SEWASIE network of mediator agents for semantic search," Journal of Universal Computer Science, vol. 13, no. 12, pp. 1936-1969, January 2007.

[15] W. Lehner, J. Albrecht, and H. Wedekind, "Normal forms for multidimensional databases," Proc. International Conference on Scientific and Statistical Database Management, IEEE Press, July 1998, pp. 63-72.

[16] C. A. Hurtado, C. Gutierrez, and A. O. Mendelzon, "Capturing summarizability with integrity constraints in OLAP," ACM Transactions on Database Systems, vol. 30, no. 3, pp. 854-886, September 2005.

[17] H. V. Jagadish, L. V. S. Lakshmanan, and D. Srivastava, "What can Hierarchies do for data warehouses," Proc. 25th International Conference on Very Large Data Bases, September 1999, pp. 530-541.

[18] L. Cabibbo and R. Torlone, "On the integration of autonomous data marts," Proc. International Conference on Scientific and Statistical Database Management, IEEE Press, July 2004, pp 223-234.

[19] L. Cabibbo and R. Torlone, "Integrating heterogeneous multidimensional databases," Proc. International Conference on Scientific and Statistical Database Management, IEEE Press, June 2005, pp. 205-214.

[20] M. Golfarelli, F. Mandreoli, W. Penzo, S. Rizzi, and E. Turricchia, "OLAP query reformulation in peer-to-peer data warehousing. Information Systems," vol. 37, no. 5, pp. 393-411, July 2012.

[21] V. Harinarayan, A. Rajaraman, and J. D. Ullman, "Implementing data cubes efficiently," Proc. ACM SIGMOD international conference on Management of data, ACM Press, June 1996, pp. 205-216.

[22] M. Rafanell and A. Shoshani, "Storm: a statistical object representation model," Proc. Statistical and Scientific Database Management, vol. 420 of Lecture Notes in Computer Science, Springer Press, 1990, pp. 14-29.

[23] M. Banek, B. Vrdoljak, A. M. Tjoa, and Z. Skocir, "Automating the schema matching process for heterogeneous data warehouses," Proc. 9th International Conference on Data Warehousing and Knowledge Discovery, Springer, 2007, pp. 45-54.

[24] S. Bergamaschi, C. Sartori, F. Guerra, and M. Orsini, "Extracting relevant attribute values for improved search," IEEE Internet Computing, vol. 11, no. 5, pp. 26-35, September 2007.