Integrating Multidimensional Information for the Benefit of Collaborative Enterprises

Marius Octavian Olaru, Maurizio Vincini
Department of Engineering "Enzo Ferrari"
University of Modena and Reggio Emilia
Via Vignolese 905b
41125 Modena, Italy
mariusoctavian.olaru@unimore.it, maurizio.vincini@unimore.it

Journal of Digital Information Management

ABSTRACT: *Collaborative business making is emerging as a possible solution for the difficulties that Small and Medium Enterprises (SMEs) are having in the current difficult economic scenarios. Collaboration, as opposed to competition, provides a competitive advantage to companies and organizations that operate in a joint business structure.*

When dealing with multiple organizations, managers must access unified strategic information obtained from the knowledge repositories of each individual organization; unfortunately, traditional Business Intelligence (BI) tools are not designed with the aim of collaboration so the task becomes difficult from a managerial, organizational and technological point of view.

To deal with this shortcoming, we provide an integration, mapping-based, methodology for heterogeneous Data Warehouses that aims at facilitating business stakeholders' access to unified strategic information obtained from a network of heterogeneous collaborating SMEs. A complete formalization, based on graph theory and the RELEVANT clustering approach is provided together with experimental evaluation of the proposed methodology over real DW instances.

Categories and Subject Descriptors:
J. [Computer Applications]; Business:
I. 2 [Artificial Intelligence]; **H.2.7 [Database Administration]:** Data Warehouse and Repository

General Terms: Data Management, Data Processing Intelligence

Keywords: Data Warehouse, Graph Theory, Data Integration, Information Exchange, Clustering Techniques

Received: 3 March 2014, Revised 19 April 2014, Accepted 28 April 2014

1. Introduction

During recent years, the economic and industrial context has seen dramatic changes that made companies undergo a series of changes in the way they perceive the business process. Some of the reasons for this phenomenon may lead back to the financial crisis, other may be the natural evolution of companies through the development of novel business visions and strategies.

The classical approach to business is characterized by the fierce competition among organizations to access assets and resources in order to produce added value that is offered to consumers. For example, one of the definitions of Economics is "*the study of the efficient allocation of scares resources among competing users*" [Casler, 1992]. This definition implies that companies are not naturally open to collaborations, especially not with similar/identical companies that have a similar activity type and offer the same kind of services, although there are some specific cases where collaboration offers an extra added value to the involved organizations (e.g., supply chains).

Large organizations usually have access to a vast array of resources and to the latest technological achievements to allow them to better adapt to market changes and to be up to date with the consumers' requirements. On the other hand, Small and Medium Enterprises (SMEs) usually struggle to compete with larger organizations whenever there is a competition for new markets and business opportunities. One way of accessing resources to be competitive is to collaborate with similar organizations that are traditionally regarded as competitors. A possible scenario could involve a network of similar SMEs working togetherto form a virtual organization that has potential access to all the resources of the involved organizations. This means creating a more active organizational structure that has higher probability to adapt to the operating scenario.

Although easy to define as a business objective, business collaboration faces some economical, business process and, ultimately, technological problems. In fact, involved companies need to deeply reason about the benefits of such an approach and need to be able to adapt their business models to fit efficiently into a collaborative environment. This means clearly defining common business objectives as well as the depth of the collaboration effort, by means of identifying the level of integration among parties. Although there is a general intuitive notion of what collaboration is, this concept is often confused with cooperation.For example, [Camarinha-Matos et al., 2009] define collaboration as a "*more demanding process (i.e. than cooperation) in which entities share information, resources and responsibility to jointly plan, implement, and evaluate a program of activities to achieve a common goal and therefore jointly generating value*", meanwhile cooperation is a similar objective that involves communication and information exchange, but for the achievement of a compatible goal.

The type of interaction among organizations thus implies a weaker or tighter coupling of the operational, managerial and strategic business processes, which in exchange determines the type and amount of interaction at operational, managerial and business level.

Furthermore, after the business objectives and business processes have been defined, companies need to be able to seamlessly integrate their information systems to exchange operational data and/or strategic information whenever the situation and the business contracts among participants may require it. One of the current limitation of collaborative business is the lack of an information infrastructure to support common decision making. In fact, nowadays companies rely on a various number of heterogeneous software tools to better manage the organization's operational and strategic activities. These software tools are produced by different software vendors and they are not entirely compatible among each other; the situation becomes even more complicated when more companies needs to exchange and integrate data, information and knowledge. In fact, information has many ways of representation; there are different paradigms, conceptual, logical and physical modelsfor storing and visualizing data and information that are often incompatible among each other. Thus, the task of creating an actual interface through which different software tools communicate may be minutely compared to the difficulties of translating information from one model to another.

Therefore, tackling these challenges and allowing companies to collaborate efficiently not only at a business level, but also at operational and technological level represents a great opportunity for companies operating in negative economic scenarios.

The purpose of this paper is to advocate the concept of collaborating business process by providing an actual integration methodology as a support for collaborating organizations. In this context we propose a three step information integration methodology for a heterogeneous Data Warehouse environment aimed at providing collaborating companies a support in their business efforts.

The rest of this paper is organized as follows: Section 2 provides an overview of related work in DW integration research while Section 3 discusses the insides of the DW integration problem and various approaches to solve it. Further, Section 4 describes a motivating example for the DW integration problem, while Section 5 provides a complete description of the DW integration methodology for heterogeneous DW environments. Furthermore, in this section there is the experimental evaluation carried out to validate the proposed methodology. Finally, Section 6 presents some conclusions of the presented work.

2. Related Work

Data Warehouse integration can be seen as a case of Data Integration, which has been an important research topic during the last decades. DW integration is a context-based DI solution where developers may exploit the available knowledge of the information to integrate. In fact, the information inside a DW is structured alongside multidimensional structures, in star-like schemas (e.g., the Dimensional Fact Model [Golfarelli et al., 1998]).

To the knowledge of the authors, there have been few attempts to formalize and solve the DW integration problem. Among them, [Kimbal and Ross, 2002] define the concept of conformed dimensions as dimensions that have consistent keys, consistent column names (i.e., dimensional attributes names), consistent attribute definitions and consistent attribute values. This is, of course, a design methodology and not a solution for a DW integration problem.

A more systematic solution is offered in [Torlone, 2008], where the author defines the dimension algebra (DA), as a set of operations (selection, projection and aggregation) that can be executed on any given dimensions. Moreover, a definition of a matching among dimensions is provided, and three properties that a matching may have: coherence, soundness and consistency. A matching that is coherent, sound and consistent is considered a perfect matching. Furthermore, the author provides a definition for compatible dimensions, with the aid of DA expressions. This approach is relevant for defining a solution from a theoretical perspective, however the author doesn't provide a method to compute the solution for actual integrated purposes, as this is outside the scope of the paper.

In [Banek et al., 2008], the authors provide a semantics-based methodology for the automated integration of heterogeneous DW schemas, based on earlier work in data integration [Bergamaschi et al., 2007; Madhavan et al., 2001]. The methodology may be regarded as an alternative to the first step in our methodology, although we believe that a systematic approach, like the one we propose, may yield better results.

In the current paper, we use a subset of the semantic mappings defined in [Golfarelli et al., 2010], in particular we used the semantic mappings for dimensional attributes: equi-level, roll-up, drill-down and related.

3. Data Warehouse Integration

Creating multi-enterprise software components or delivering methods to integrate the already operational ones canprovide a strong incentive for collaborating businesses. Such a goal, however, requires identifying and tackling a series of technological shortcomings;in particular the entire process must follow these steps:

1. Analyze the business strategy and identify the software tools that need to cooperate;

2. Analyze each individual tool and their internal and external model for storing/representing data, information and knowledge;

3. Develop interfaces that allow the exchange of the information among various tools/models;

The level of interoperation determines the amount of data and information that the organizations need to exchange, as well as the granularity. It may go from raw operational data (like specific product/service information, e.g. creation date, location at the current time, etc.) to high-end, aggregated strategic information that managers use for taking business decisions. In any kind of collaboration, however, companies inside the network need to have a global overview of the entire business structure, and managers need to be able to take strategic decisions based on information coming from all the companies. Traditionally, Business Intelligence tools are used for this purpose, mainly a Data Warehouse. The general notion about the importance of the DW for the decision making process inside organizations makes it the primary target for integration purposes. In fact, collaborating companies need to be able to integrate strategic information obtained from each of the companies' information repository, for the benefit of the decision making of the entire business network.

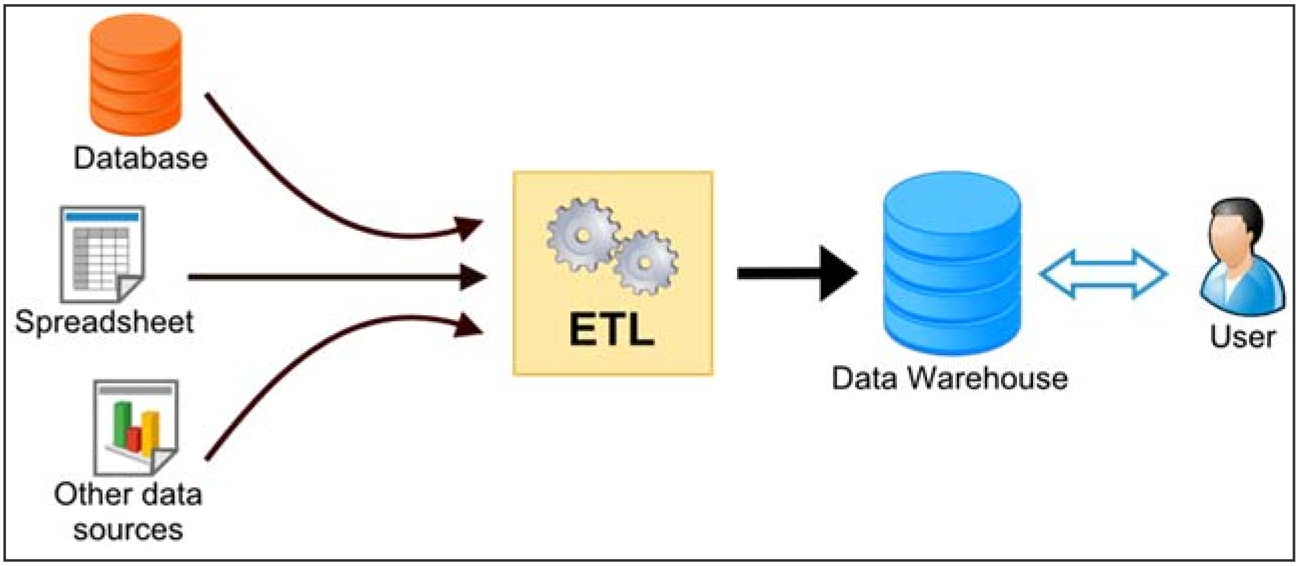

Figure 1. Classical DW Architecture

3.1 Data integration in a DW

A Data Warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision-making process [Inmon, 1992].

A part from the other properties cited in the definition, we point out that a DW is traditionally focused on one single company and it usually integrates and reconciles a different number of heterogeneous data sources (Figure 1), that are found out across the entire company. The data sources may vary from flat files, spreadsheets, semi-structured data (like XML) to Database Management Systems (DBMSs) and web data [Calvanese et al., 2001].

Even after identifying the correct data sources to integrate for building the DW, the actual integration task itself implies resolving a series of schema and instance level conflicts. In fact, the different data sources may contain the same information, but structured differently (e.g., different classes/entities/tables, etc.); even if the schema is very similar or identical, the instance values may differ (e.g., different names for the same real-life concept, different date format, abbreviations, etc.). Data coming from the data sources undergoes a series of changes and transformations before being loaded in the final Data Warehouse, through a series of operations called Extract-

Transform-Load (ETL). This approach, however, is laborious, time consuming and mostly manual. It thus implies high costs for the companies.

3.2 High end DW integration

The DW creation and load procedure produces valuable information that among other characteristics should be available with the smallest delay and should be of elevated quality. When integrating information obtained from multiple DWs, one simple solution deriving from classical Data Warehousing would be to use *extensive* ETL procedures to combine all the data sources used to populate each and one of the individual DWs, into a unique data repository, usually called *data staging area*, and on top of that build a new DW.

We envision instead a more direct integration methodology where it is possible to combine the high-level information already available in the final DWs in a seamlessly way, allowing the end-users to benefit from a number of advantages, like data quality, business network flexibility and development time reduction.

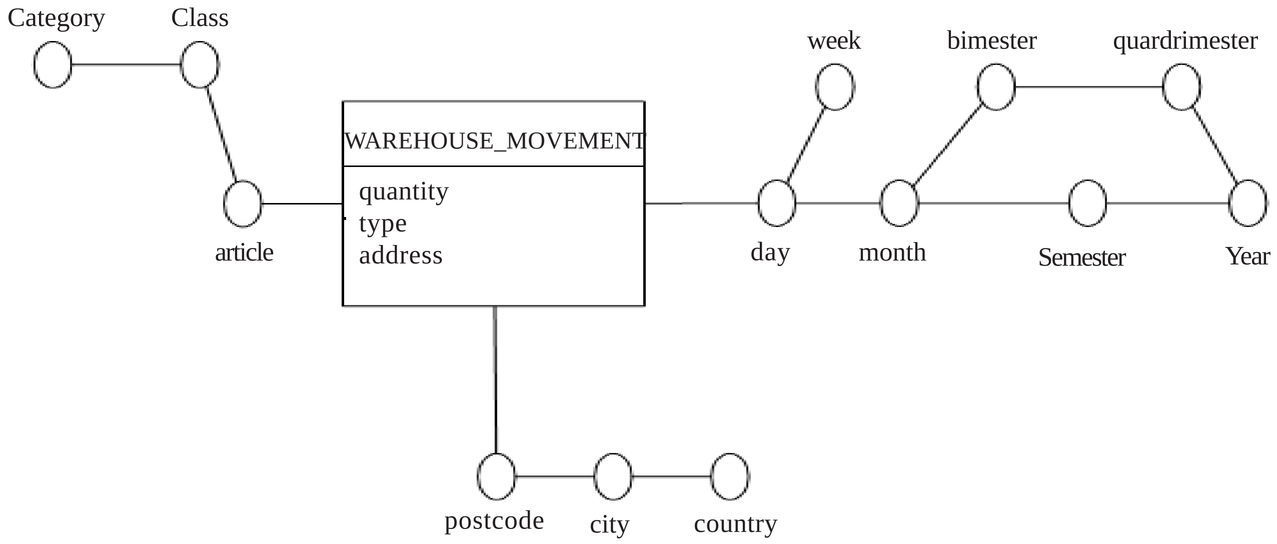

Figure 2. Example DW1

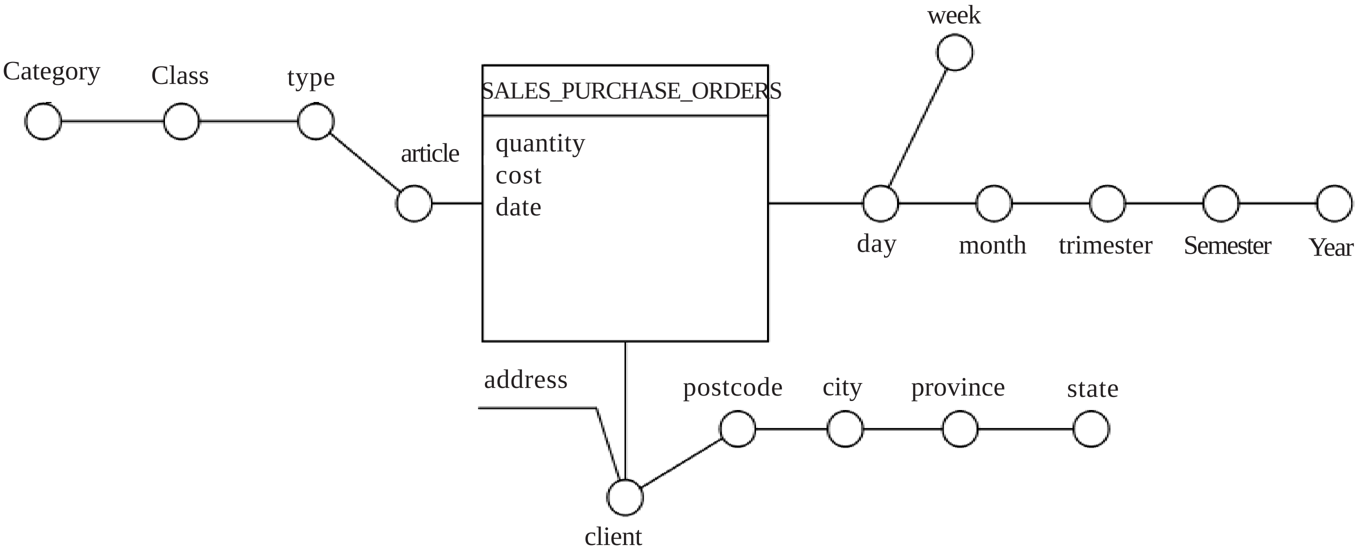

Figure 3. Example DW2

4. Motivating example

In a real-case DW integration scenario, developers must face two types of problems. First, the conceptual schemas are usually different. Consider, for example, that Figure 2 and Figure 3 represent two DW belonging to two different real companies, the warehouse movements DW and the purchase orders DW. Although belonging to different companies and built for different purposes, the DWs present similar dimensions: the time hierarchy, the geographical destination and the class hierarchy. Schema inconsistencies are present in all three dimensions, but the fact that they are inconsistent does not mean that the dimensions are incompatible, but only that the same information is represented differently. In Figure 2, for example, a month is aggregated in bimesters and quadrimesters and finally in years (or alternatively in semesters and years), while in the second DW a month is only aggregated in semesters and years. In this case, the first warehouse has additional information with respect to the second warehouse. The second possible inconsistency is the instance level inconsistency, which means representing the same information according to the same conceptual schema, but using different attribute values. Different working groups may represent the same information differently. For example, the name of the months may be represented using the full name in one DW (e.g., January, February, etc.) or may be abbreviated in another DW (e.g., Jan., Feb., etc.). Among two distinct, heterogeneous DWs there may be both inconsistencies, and information integration tools must be aware of them, and correctly identify the same relevant information, and not to consider it independently.

5. DW Integration methodology

Data Warehouses integration can be seen as a subcase of classical data integration (DI), where developers usually make use of mappings to express semantic similarities among similar attributes/concepts [Bergamaschi et al, 1998, Beneventano et al., 2001]. The mappings are reused by query rewriting techniques to execute a global query over the local data sources and to integrate and reconcile data coming from different, compatible data sources. The experience accumulated in the two decades of DI research may also be used in DW integration, where similar mapping-based techniques may be used.

In this context, we propose a three step DW dimension integration methodology that is able to:

1. Identify similar dimensional attributes and generate semantic mappings;

2. Integrate and extend compatible dimension hierarchies;

3. Populate the imported dimension attributes with relevant information;

5.1 Generate semantic mappings

In this section we present an instance level [Rahm, 2001] technique to automatically generate semantic mappings among different DW dimensional attributes. We rely on cardinality based properties, in particular on a graph-like structure that can be extracted to represent the different dimensional hierarchies of the two DWs, called cardinality-ratio[Bergamaschi, 2012a].

Consider, for example, that the time dimension on the first DW (Figure 2) contains all the days and months from January 1st 2001 to December 31st 2012 (12 complete years), the time dimension of the second DW (Figure 3) contains all the days from January 1st 2008 to December 31st 2013 (16 complete years), and that there is an inconsistency among the instances of the schemas, such that an intersection of the attribute values of the schemas is null. Although apparently the schema and the instances are different, they represent the same real - world concept, and some properties deriving from the general knowledge of that concept is maintained. For example, in both dimensions, a month is an aggregation of 30-31 days (i.e.,

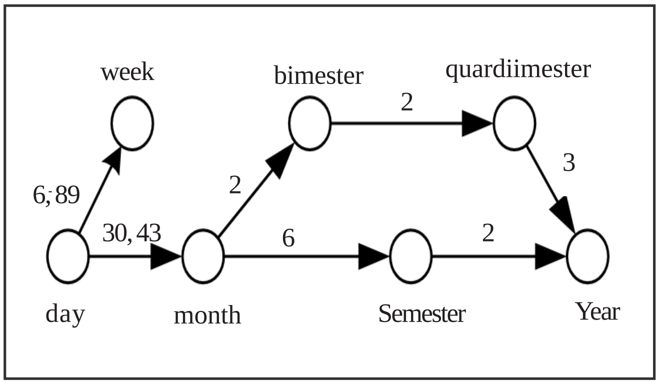

Figure 4. Graph 1

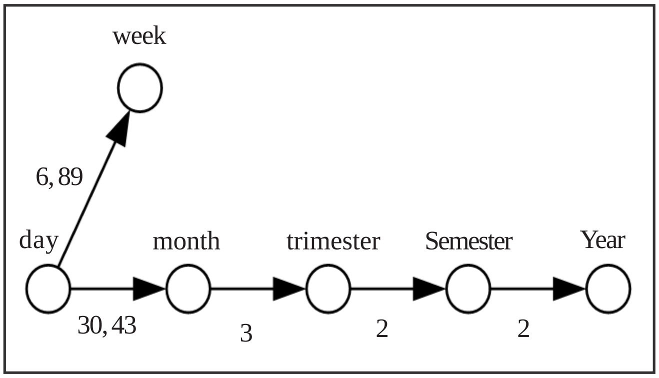

Figure 5. Graph 2

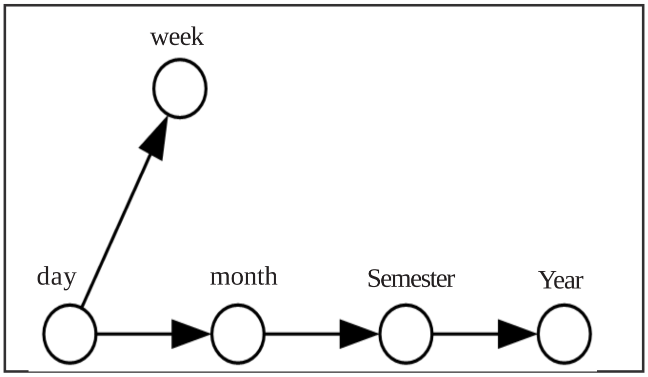

Figure 6. Common Subgraph

the ratio among the number of distinct values of the attribute month and the number of distinct values of the attribute day), also a semester is an aggregation of 6 different months in both dimensions. This common information may be used to generate semantic mappings among similar dimension hierarchies.

In particular, we consider the dimension hierarchies as directed labeled graphs, where the label of each graph is the cardinality ratio among different dimension hierarchies (Figures 4 & 5). A connectivity matrix (Figure 7) is associated to each graph, each element of the matrix a_{ij} is greater than 0 if there is a path among the two nodes. The value of each element a_{ij} is the cardinality ratio among the two dimensional attributes that the nodes represent. Such cardinality ratio is computed as the multiplication of the labels of the arcs the path is composed of. In the example above, the lines of the connectivity matrix M_1 are assigned to the nodes day, week, month, bimester, quadrimester, semester and year respectively. Thus, the element $a_{34} = 2$, as a bimester is composed on average of two months. Similarly, the matrix M_2 is assigned to the second graph, and the lines correspond to the nodes day, week, month, trimester, semester and year respectively.

We then use Algorithm 1[1] to compute a common subgraph that is used to generate the semantic mappings. The algorithm, given an error $\in$, computes a maximum rank common square sub-matrix, whose element $a_{i,j}$ has a relative error of no more than $\in$ from the average of the same elements in the initial matrices. The new matrix describes a subgraph, that approximates the two initial graphs with respect to the cardinality ratios by no more than a relative error of $\in$.

In the example above, similar names were used for the dimensional attributes/nodes, so that it is easier to visualize the kind of semantic correspondences that are maintained between the two initial graphs and the common subgraph. In real life cases, however, the attribute names are usually different, abbreviated, or are just labels with no particular meaning. In such cases, it is difficult to identify similar/identical dimensional attributes.

$$M_1 = \begin{pmatrix} 1 & 6,99 & 30,43 & 60,86 & 121,72 & 182,58 & 365,16 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 4 & 6 & 12 \\ 0 & 0 & 0 & 1 & 2 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_2 = \begin{pmatrix} 1 & 6,89 & 30,43 & 91,29 & 182,58 & 365,16 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 3 & 6 & 12 \\ 0 & 0 & 0 & 1 & 2 & 4 \\ 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Algorithm 1: Cardinality Ratio Computation

```
C = {empty matrix}
for every square sub-matrix $M_1$ of the first matrix do
for every square sub-matrix $M_2$ of the second matrix do
 if for every
 $i, j; m_{ij}, m_{ij} \in [(1-\in)\frac{|a_{ij}-m_{ij}|}{2}, (1+\in)\frac{|a_{ij}-m_{ij}|}{2}]$ then
 if $ranks(S_A) > rank(C)$ then
 $C =$ new matrix of $rank(S_A)$
 for every $C_{ij}$ do
                  $C_{ij} = \frac{|a_{ij}-m_{ij}|}{2}]$
                      end for
                    end if
                end if
            end for
       end for
 return C
```

The node day of the common subgraph is obtained from the node day of the first graph and from the node day of the second graph. This means that the two initial nodes are the same. Similarly, the corresponding nodes week, month, semester and year are the same. Starting from these nodes, it is possible to generate semantic mappings between all the other dimensional attributes.

As mapping predicates, we used those defined in [Golfarelli et al., 2012], in particular the equi-level, drill-down, roll-up and related semantic predicates.

To generate semantic mappings, we use the following inference rules:

Let A, B be nodes of the first graph, and X, Y be nodes of the second graph, such that B is an aggregation of A and Y an aggregation of X.

Rule 1. If A and X correspond to the same node in the subgraph, then A {equi-level} B.

Rule 2. If from Rule 1, B {equi-level} X, then A {drill-down} X and X {roll-up} A (Figure 8).

Rule 3. If from Rule 1, A {equi-level} X, then B {roll-up} X and X {drill-down} B (Figure 9).

Rule 4. If from Rule 1, B {equi-level} X, then A {roll-up} Y and Y {drill-down} A (Figure 10).

[1] The algorithm s heuristic, with a computational complexity of at least $O(n^4)$, where n is the minimum number of categories in either one of the two considered dimensions. Given that the number of categories in one dimension is fairly limited (thus the computational complexity is limited), any further optimization is outside the scope of this paper.

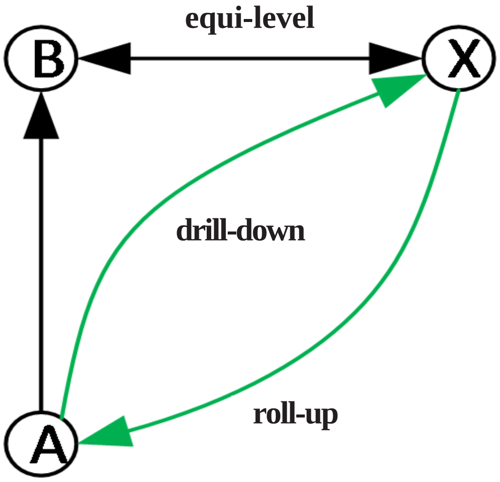

Figure 8. Rule 4

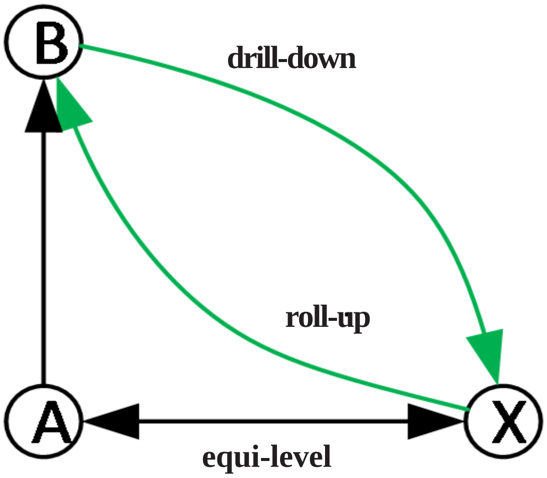

Figure 9. Rule 3

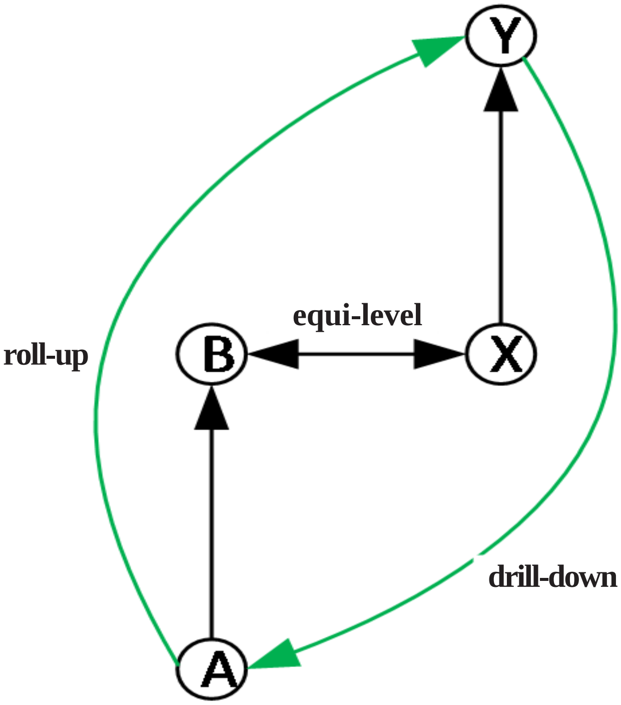

Figure 10. Rule 3

Rule 5. Add a {related} predicate for each other pair of nodes.

5.2 Experimental evaluation

In order to validate the mapping discovery approach, the matching technique was testedon a scenario of three independent DWs containing sales data of three different companies (an automation technology company, a food producer and a fashion designer). The three DWs contain almost 250.000 purchase orders combined, all related to approximately 9.000 customers.

The dimensions of the DWs vary from 4 to 10 dimension hierarchies each, for a total of around 50 dimension hierarchies for each DW. The hierarchies were represented in denormalized relational form, in standard industry DBMSs.

The first step of the validation was the manual annotation of the dimension hierarchies into the dimension graphs that were subsequently used for the mapping algorithm. Pairs of similar dimensions were first manually matched, and subsequently matched using the proposed matching algorithm.

To measure the mapping results ε, we use the maximum error ε, i.e. the maximum deviation between two corresponding elements of the connectivity matrices describing the two analyzed dimensions, from their mean value. A low error ε is indication that the computed mappings connect two dimension hierarchies that have similar nodes and that are connected in a similar manner.

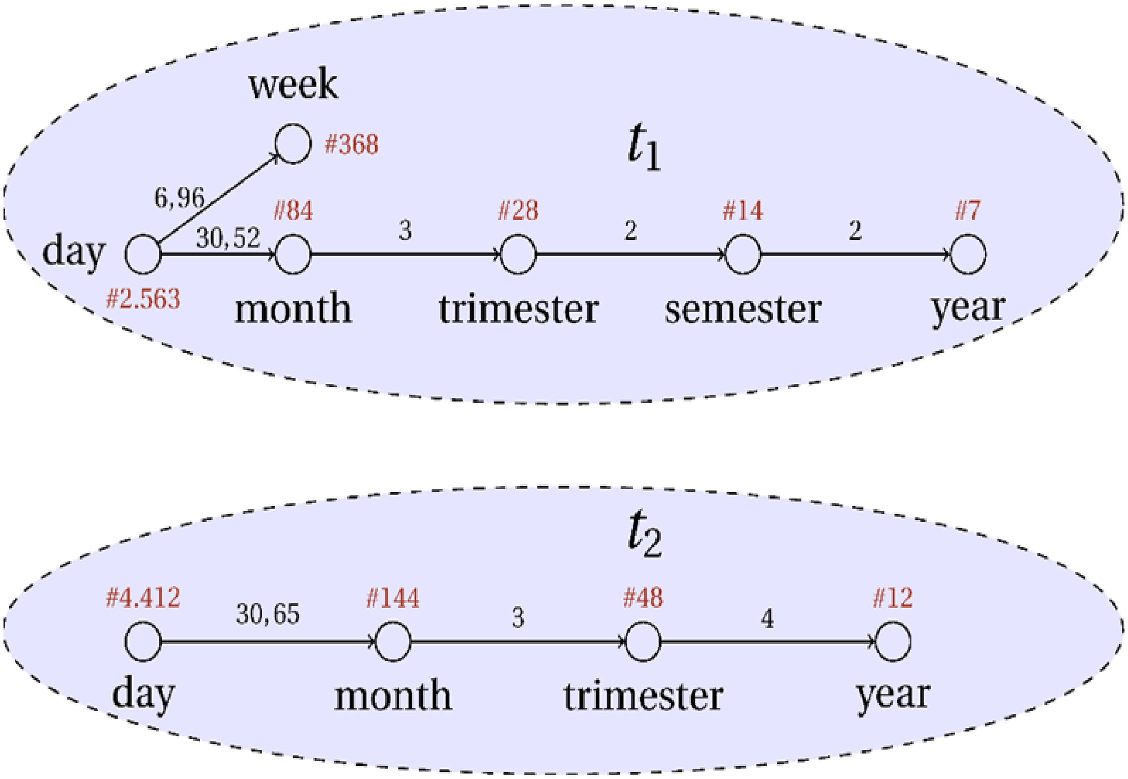

Figure 11. Time dimensions

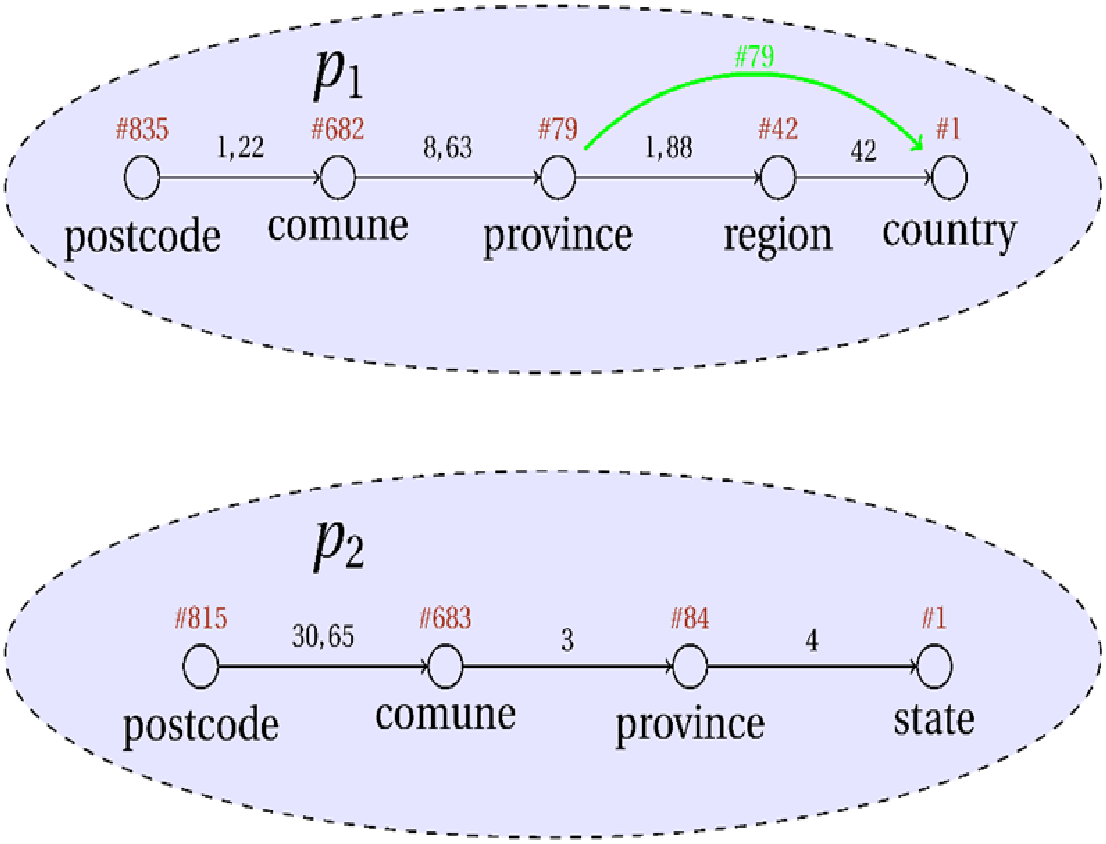

Figure 12. Postcode Dimensions

The first test involved time dimensions. The expectation was that the common dimensions matched with a low error ε because time hierarchies generally have a fixed structure over all DWs. Figure 11 contains the description of two of the three time dimensions, with the total number of distinct members of each category (red numbers) as well as the cardinality ratios between various aggregation levels (numbers above the edges). The instance data of the schemas (the sets of members of the base categories)was 75% overlapping, and the common submatrix used to compute the common subhierarchy contained an error $\varepsilon = 4\%$.

The second test involved a geographical hierarchy (Figure 12 contains two of the three dimensions). Unlike the time hierarchy, the probability of having the same structure is lower; however some similarities are maintained. A common subhierarchy was computed from the two original dimensions with an error of ε = 7,4%. Furthermore, the intersection of the members set was analyzed, and was discovered that the instances contained only 16% common data (the other values were totally distinct). This also serves as a proof that a value-matching approach would consider the two dimensions with a low similarity, although the dimensions are conceptually identical.

postcode		article		day	
overlapping	ε	overlapping	ε	overlapping	ε
16 %	12 %	0 %	4 %	75 %	0,1 %

Table 1. Evaluation results

The final test regarded the article dimension hierarchy. As expected, the sets of members of the dimension categories are completely disjoint because the companies sell different product/services. However, the mapping discovery methodology is still able to compute a common subgraph with an error of ε = 16%, and subsequently to correctly identify and map related categories.

Table 1 synthesize the results of the experimental evaluation. As can be easily noted, independently of the common data that the dimensions share, the proposed methodology is able to correctly compute semantic mappings between dimension hierarchies with a worse case error rate of only 16%.

5.3 Schema level integration

The semantic mappings can be exploited to write and execute, where possible, queries over the two instances. The expressiveness of the queries is, however, limited to their compatibility, which means that some queries are simply impossible to execute on both instances, due to their schema level inconsistency; for example, the fact measures are not aggregable by bimester in the second instance of the given example, as the required information is not available.

A DW integration methodology must be general enough to be efficient in a variety of integration architectures, either involving two or more DWs. Some integration architectures, like the Peer-to-Peer DW [Golfarelli et al., 2012, Beneventano et al., 2005] hypothesizes a network composed of a potentially large number of peers, each having its independent DW. In such scenario, the possibility of rewriting queries on all the peers is drastically reduced by the compatibility of the schemas. Some workarounds may include the possibility to deny the queries that are incompatible with all the nodes or to execute each query only on compatible nodes. This kind of approaches, however, may create confusion rather than provide a solution for the problem, as for some cases the end user may receive a unified answer obtained from all the nodes of the network, a part of the network, or may obtain data belonging only to the local node.

A possible solution may be to uniform similar dimensions, by importing, where possible, remote compatible dimension levels. For example, the time dimension in the first schema contains the bimester-quadrimester alternative path that is not available in the second schema, meaning that the users of the first schema have augmented analysis and interpretation possibilities compared to the users of the second schema. If, however,

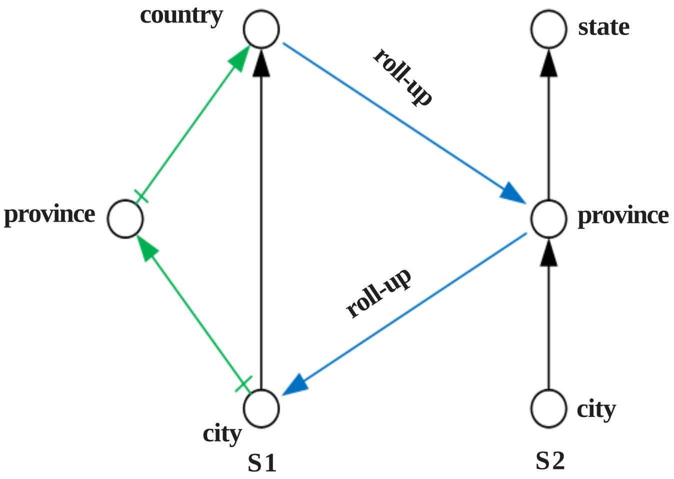

Figure 13. Importation rule, example

there is sufficient knowledge about the two schemas, the dimensional attributes may be imported in the remote schema, uniforming the users' capabilities to jointly interrogate the information repositories, in the same way. The schema knowledge in this case may be the mappings discovered in the first step of the presented methodology, that provide indication of how the dimensional attributes fit in the other schemas. This way, the missing attributes may be seamlessly integrated in the remote schema, by also generating the correct semantic relationships relatively to the other dimensional attributes available in the schema. For example, in the schemas in Figure 2 & 3, the province dimensional attribute was available only in the second schema, but can be also imported in the other schema (see Figure 13). From the first step, the mapping S2.province {roll-up} S1.city was derived from Rule 2 (blue line), so the province attribute is inserted in the schema S2 as an aggregation of city. Also, S1.country {roll-up} S2.province, so country is an aggregation or province in S1.

The reason the attribute was inserted as optional is that there may be the case that some cities in the instance of the first schema don't have a region, or none is deductible from the instance of the second schema. The concept will be explained in the next value importation section.

5.4 Importing values with RELEVANT

One important step in the DW integration procedure is the integration and reconciliation of common information. This implies that: (1) the newly imported attributes have to be populated with consistent values and (2) the possible inconsistencies among attribute values must be resolved. For this purpose, we propose an extension of RELEVANT (RELEvantVAluesgeNeraTor) [Bergamaschi et al., 2007] that is specially conceived for the integration of multidimensional information.

RELEVANT performs clustering of attribute values and, for each cluster, identifies one relevant value, that is representative for the whole cluster. By applying the RELEVANT techniques to the values of the dimensional attributes, we obtain clusters of related dimension. The relevant value provided by each cluster is then used for populating the missing values of the newly imported dimensional attributes. In this way, RELEVANT is able to provide approximate values to the new dimension attributes.

Clusters of related elements are computed by using some similarity measures. In this extension of RELEVANT, clusters are created by means of two similarity measures: 1. syntactic similarity, which compares the alphabets used for describing the attribute values; 2. containment, which measures the closeness of attributes belonging to different dimension. In particular, the containment is based on the {roll-up}, {drill-down}, and {equi-level} mappings holding among the sources. Some further semantic measures based on lexical similarity and external ontologies can be exploited for dimensions belonging to specific domains, but this is outside the scope of the paper.

Example. Consider the attribute importation in the earlier example. For a more realistic example, consider that the newly inserted attribute must be populated using integrated information obtained from other three distinct information sources (Table 2) that contain related values.

The way we applied RELEVANT is similar to functional dependency reasoning, but rather than checking if a certain functional dependency holds, we enforce the one given by the roll-up relation, not on each individual values but rather on cluster of similar values. In other words, we impose that if two cities belong to the same cluster, then they must belong to the same cluster of region values. For example, the values "*Torino*" and "*Turin*" are in the same cluster, so the values of the region value must also belong in the same region. In the last instance (red values) there is no region specified, so the city "*Turin*" is assigned the representative value for the corresponding region cluster (Figure 14). Similarly, the city "*Bologna*" (red section in the table) is placed in the region "*Emilia Romagna*", which is the relevant value of its cluster.

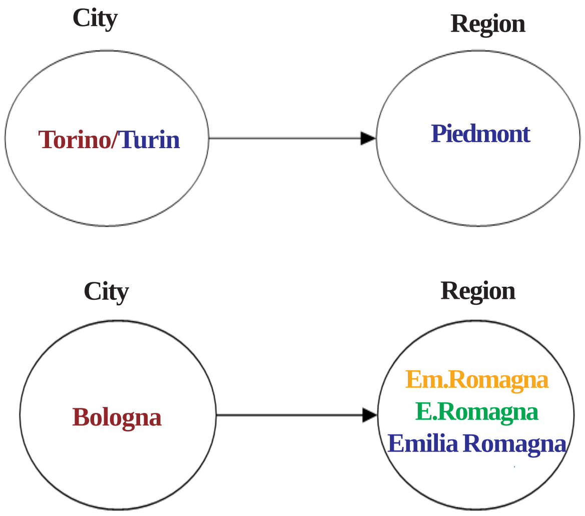

Figure 14. Example Clusters

City	Region
Roma	Lazio
Firenze	Toscana
Ferrara	Em.Romagna
Milano	Lombardia
Palermo	Sicilia
Palermo	Sicilia
Milano	Lombardia
Pisa	Toscana
Rimini	E. Romagna
Palermo	Sicilia

City	Region
Florence	Tuscany
Bologna	Emilia Romagna
Rome	Lazio
Turin	Piedmont
Palermo	Piedmont
Rome	null
Torino	null
Bologna	null
Palermo	null
Firenze	null

Table 2. Example values

Conflicts resolution

One of the advantages of applying RELEVANT is the possibility to resolve instance level inconsistencies. In fact, having multiple information sources allows analysts to discriminate between allowed or incorrect attribute values in the given instances. For example, the regions of the city "*Sicilia*" must be in the same cluster, which is not the case in the example in Table 2. RELEVANT is able to detect that the region values do not belong in the same cluster, so the last one is discarded (Figure 15). In such cases, the ability to discriminate among correct or wrong information is given by the number of values in the cluster. In this case, there are three values, two of which are correct, so a simple majority approach may be applied. In other cases, however, there may not be sufficient available information to be able to choose the right value. In any case, given the quality requirements in a DW we suggest human supervision in such scenarios.

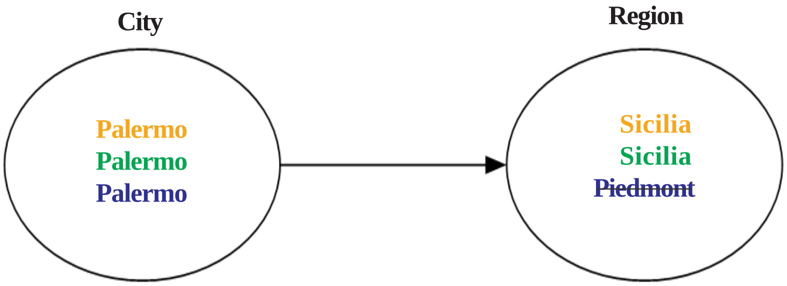

Figure 15. Example Clusters

6. Conclusions and Future Work

As the economical context is evolving, DW integration will become an even more demanding challenge and managers and developers will seek new and innovative ways to obtain relevant information from different sources. Latency is already an issue in today's Data Warehousing, as the amount of data in a warehouse builds up incrementally; that is why combining information from multiple information repositories not only increases the development and execution time, but also increases the latency of data almost exponentially. Managers nowadays and in the future will require near real-time information, so the approach proposed in this paper may seem an obvious solution to some of the problems of the current BI approaches.

The work presented in this paper brought two general conclusions.

First of all, when dealing with collaborative business making, a clear distinction must be made between the dynamics of the business process and the integration methodology that is adopted. The business process dictates the sequence and type of interactions among parties, and the type of information and knowledge to be exchanged; meanwhile the methodology defines the technical background that the exchange process is operating in. The current paper defines a three step integration methodology that should be general enough to fit in any kind of collaborative business structure.

The second conclusion is that, although works fine when dealing with a limited number of sources, the current methodology shows its great potential when tackling a scenario with numerous organizations each having its own DW. In fact, introducing an automated methodology has the potential of drastically reducing the required development and execution of the integration phase, providing near real-time strategic information for managers.

For the future, we plan to analyze different mapping methodologies (for example, semantics) and to study and test combined mapping generation methodologies in order to increase the accuracy of the mappings.

Furthermore, a study among the multidimensional to relational schema transformations will be conducted in order to allow the design of a complete, fully-working prototype.

Bibliography

[1] Banek et al., Banek, M., Vrdoljak, B., Tjoa, A. M., Skocirm, Z. (2008). Automated Integration of Heterogeneous Data Warehouse Schemas. *IJDWM*, 4 (4) 1–21.

[2] Beneventano et al., D. Beneventano; S. Bergamaschi; F. Guerra; M. Vincini. (2001). The Momis approach to Information Integration - Third International Conference on Enterprise Information Systems - ICEIS Press Setubal (PRT)) - n. 1.

[3] Beneventano et al., Beneventano, D., Bergamaschi, S., Guerra, F., Vincini, M. (2005). Querying a super-peer in a schema-based super-peer network - International Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P 2005) - Springer, Lecture Notes in Computer Science Berlino (DEU)).

[4] Bergamaschi et al, Bergamaschi, S., Castano, S. De Capitani De Vimercati; S., Montanari; S., Vincini, M. (1998). An Intelligent Approach to Information Integration

- 1st Conference on Formal Ontology in Information Systems (FOIS ’98)

[5] Bergamaschi et al., Bergamaschi, S., Sartori, C., Guerra, F., Orsini, M. (2007). Extracting Relevant Attribute Values for Improved Search. *IEEE Internet Computing*, 11 (5) 26–35.

[6] Bergamaschi et al., S. Bergamaschi, S., Bourquet, P., Giacomuzzi, D., Guerra, P., Po, L., Vincini, M. (2007b). MELIS: an incremental method for the lexical annotation of domain ontologies. *International Journal on Semantic Web and Information Systems* (IJSWIS) 3 (3) 57-80.

[7] Bergamaschi et al., Sonia Bergamaschi, Francesco Guerra, MirkoOrsini, Claudio Sartori, Maurizio Vincini. (2011). A semantic approach to ETL technologies, in *Journal of Data and Knowledge Engineering*, 70 (8), August p. 717-731.

[8] Bergamaschi, Bergamaschi, S., Olaru, M. -O., Sorrentino, S., Vincini, M. (2012). Dimension matching in Peer-to-Peer Data Warehousing. DSS (p. 149–160).

[9] Calvanese et al., Diego Calvanese, SilvanaCastano, Francesco Guerra, DomenicoLembo, Michele Melchiori, Giorgio Terracina, DomenicoUrsino, Maurizio Vincini. (2001). Towards a Comprehensive Methodological Framework for Integration - Proceedings of the 8th International Workshop on Knowledge Representation meets Databases (KRDB 2001)

[10] Camarinha-Matos, Luis M. Camarinha-Matos and HamidehAfsarmanesh. (2006). Collaborative Networks: Value creation in a knowledge society - in Knowledge Enterprise: Intelligent Strategies in Product Design, Manufacturing, and Management, *In*: Proceedings of PROLAMAT, IFIP TC5 International Conference, June 15–17, Shanghai, China

[11] Camarinha-Matos et al., Luis M. Camarinha-Matos, HamidehAfsarmanesh, Nathalie Galeano, Arturo Molina. (2009). Collaborative networked organizations – Concepts and practice in manufacturing enterprises, *Journal of Computers & Industrial Engineering*, 57 (1) 46-60, August.

[12] Casler, Stephen Casler, *Introduction to Economics* (New York: HarperCollins. p. 3 [Guerra et al., (2012). Francesco Guerra, Marius-Octavian Olaru, Maurizio Vincini (2012) - Mapping and Integration of Dimensional Attributes Using Clustering Techniques. - Lecture Notes in Business Information Processing -Springer New York (USA)) - n. 123.

[13] Golfarelli et al., Matteo Golfarelli, Dario Maio, StefanoRizzi: The Dimensional Fact Model: A Conceptual Model for Data Warehouses. *Int. J. Cooperative Inf. Syst.* (IJCIS), 7 (2-3) 215-247.

[14] Golfarelli et al., M. Golfarelli, F. Mandreoli, W. Penzo, S. Rizzi, E. Turricchia. Towards OLAP query reformulation in Peer-to-Peer Data Warehousing. *In*: I.-Y. Song and C. Ordonez, editors, DOLAP, p. 37–44. ACM.

[15] Golfarelli et al., Golfarelli, M., Mandreoli, F., Penzo, W., Rizzi, S., Turricchia, E. (2012). OLAP Query Reformulation in Peer-to-Peer Data Warehousing. *Information Systems*, 37 (5) 393-311, July.

[16] Inmon, William H. Inmon. (1996). Building the Data Warehouse, Ed. John Wiley & Sons, Inc. New York, NY, USA.

[17] Kimbal, Ross, R. Kimball, M. Ross. (2002). The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. John Wiley & Sons, Inc., New York, NY, USA, 2nd edition.

[18] Madhavan et al., J., Madhavan, P. A. Bernstein, E. Rahm. Generic Schema Matching With Cupid. In: P. M. G. Apers, VLDB, p. 49–58. Morgan Kaufmann.

[19] Rahm, E., Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, 10 (4) 334–350.

[20] Torlone, R. (2008). Two Approaches to The Integration of Heterogeneous Data Warehouses. *Distributed and Parallel Databases*, 23 (1) 69–97.