Coopetitive Data Warehouse: A Case Study

Andrea Maurino, Claudio Venturini, and Gianluigi Viscusi

University of Milano Bicocca, viale sarca 336, edificio U14, Milano, Italy
{maurino,venturini,viscusi}@disco.unimib.it

Abstract. In this paper we discuss the experience of the development of a real system for integrating data about turnover, price and selling volume of AOP UnoLombardia, the biggest association of fruit and vegetable growers in the Lombardia region (Italy), that includes primary Italian and European brands such as Bonduelle and Dimmidisi. The system represents an adaptation and transformation of traditional data warehouse repository oriented development to comply the requirements of a coopetitive environment, where multiple organizations are willing to cooperate over some topics but, at the same time, they compete in the market. Readers may found useful insights and lessons learned from the following contributions of the present work: (i) a methodology to design data warehouse applications in a coopetitive environment and (ii) an architecture based on the combination of virtual data integration and traditional ETL enforcing protection of sensible data.

Keywords: Coopetition, Data Warehouse, Case Study, Agrifood chain.

1 Introduction

Coopetition [1] is a kind of relationship between several firms that expose, at the same time, a cooperative and a competitive behavior. Although the topic is well studied in the fields of economy and organization management [2], the effects of coopetition on information systems design planning hasn't been analyzed in enough detail, apart from some contributions on factors enabling knowledge management and sharing [3, 4]. In this paper we focus on information sharing and data integration, discussing the drivers of information systems design and development in coopetitive environment. In particular, we focus on data warehouse as a specific information system and one of the most diffused approach to data integration. The need of data warehouse applications can be found in almost all organizations of companies when there is the need to put together information related to the their market [5]. The main issue data warehouse has to solve is the one of correctly and efficiently merge data sets from multiple, autonomous, heterogeneous data sources (the so called local data) into a unique data set (the so called global data), that can be queried according to some business dimensions. All researches related to the field of data warehouse start with the basic assumption that data sources can be accessed without any business limitation. In coopetitive environments this assumption is not always verified

C. Salinesi, M.C. Norrie, and O. Pastor (Eds.): CAiSE 2013, LNCS 7908, pp. 482–497, 2013.

due to the fact that part of the information to be shared (e.g. the price of goods, the name of customers) represents core business data and thus it is a sensitive information. However in some real situations, there is the strategic need to put such sensitive data into a data warehouse application to obtain a better vision of the whole market. Taking these issues into account, we define a coopetitive data warehouse as a data warehouse where there is the need to a real integration of business data coming from the day-by-day activity provided by organizations exposing a coopetitive behaviour. As a consequence of this definition, a coopetitive data warehouse development it is worth to be investigated in its differences and similarities from traditional data warehouse, if any. In this paper we aim to provide a contribution to the field through the description of the case of the design and development of a coopetitive data warehouse, considering a software architecture based on the combination of virtual data integration and traditional ETL techniques to enforce the protection of sensitive data. The research method adopted follows design science paradigm [6], carefully combined with an action research perspective [7], due to the relevance of the relationship with an institutional client. In particular, as for evaluation issues [6], we explore a case study that reports the experience in the design and development of a coopetitive data warehouse application for AOP UnoLombardia, the main association of fruit and vegetables growers in the Lombardy region (Italy), whose members asked for a solution allowing a more complete vision of their market to define common strategies with regard to their customers, that are national and international large-scale retails.

The paper is organized as follows: Section 2 provides the theoretical background describing the coopetitive behaviour by means of the game theory. Its results provide us specific drivers as high level requirements related to the development of a coopetitive data warehouse. Section 3 describes the methodology we defined and used to support the development of the coopetitive data warehouse. Section 4 discusses the application of the proposed methodology to the case of AOP UnoLombardia, while Section 4.1 describes the resulting software architecture we designed and developed, and Section 5 reports the evaluation of the system in the real case. Finally Section 6 draws the conclusions and future works.

2 Theoretical Background

In this Section we aim to provide the theoretical background to identify the specific high level requirements of a coopetitive data warehouse. It is worth noting that these requirements are on the one hand necessary, being related to the conditions under which a company may share reserved information to competitors, but on the other hand, they are not sufficient, due to the need to elicit the requirements of each different context where a coopetitive data warehouse is supposed to be developed. As for the identification of the conditions under which a company may share reserved information to competitors, we formalize coopetition by means of game theory [8],[9],[10]. Let A and B be two companies

and let $\{share\ (S), not-share\ (NS)\}$ be the two possible actions to share some information or not. Values k_A and k_B quantify the amount of knowledge shared by A and B respectively, and they can be estimated, for example, associating the revenues related to the data shared. Nevertheless, when information leaves the border of the firm, it faces a potential loss related to possible disclosures of data. Thus data sharing is associated to a loss value lv[1]. The advantage related to data sharing can be described by a utility function $av = f(k_a, k_b)$ that estimates how useful is the integration of data. Let assume that $av >= 0$, $lv >= 0$, $k_A >= 0$ and $k_B >= 0$, we can model the coopetitive game by means of the matrix represented by Table 1.

Table 1. coopetition matrix

		B	
		not share	share
A	not share	(k_A, k_B)	$(k_A, k_B - lv)$
	share	$(k_A - lv, k_B)$	$(k_A - lv + av, k_B - lv + av)$

Assuming that A and B are rational players, we can see that if $av <= lv$ then the strategy of sharing is dominated by the strategy of not sharing. A generic firm x could be interested in joining the network if and only if the utility related to sharing its data $(k_x - lv + av)$ is more than the utility of not sharing (k_x). That is when:

$$av > lv \tag{1}$$

According to equation 1 the condition for sharing data of a firm is that the estimated value for aggregated information is more than the one related to the loss of exposed information. Notwithstanding the vast literature on information utility and value [11–17], to the best of our knowledge there is no agreed way to formally define av and lv. Nevertheless, in the following we rely on the main issues emerging from the literature on information utility and value, *access* [18], *trust* [19] as related to *privacy* [20, 21], and *quality* [15, 17] for first describing the requirements that influence lv; whereas requirements influencing av are the same of traditional data warehouse applications. Moreover it is worth nothing that sharing sensitive data is mainly based on the trust that a single company has with respect to the organization; thus av includes also the value to be member of an organization. According to the above mentioned literature results, the most relevant requirements that affects lv are i) *access control policies*; ii) *privacy preserving strategies*; iii) *quality of shared data*. We now discuss the motivations for the relevance of these requirements.

As for *access control policies*, they directly impacts information loss value due to the fact that lv is directly related to the probability of exposing sensitive data

[1] We do not distinguish between the loss value of A and B because this does not affect the outcome of the game.

to competitors. As a consequence there is the need to pay special attention to the requirements related to access control policies in the design of the coopetitive data warehouse application.

In particular, considering the information flow in a coopetitive data warehouse, three are the phases where it is mandatory to define strong policies for access control:

- during the extraction of data from organizations
- during their integration and loading into the DW
- during the presentation of results to organizations that are members of the coopetitive network or to outside actors.

Privacy preserving strategies are another important issue in the design of a coopetitive data warehouse application, because they must guarantee that no data mining technique can exploit the global integrated data to infer knowledge to any single firm. The problem of preserving privacy trough data mining techniques is well studied in the literature [22], but in general proposed solutions are tailored against specific types of attack. This implies that to design an effective privacy preserving mechanism, there is the need to define a correct model of the possible attacks. A coopetitive data warehouse application, like a traditional data warehouse, should allow the users to query the data in a highly flexible way. In other words a coopetitive data warehouse application doesn't bound the user to a set of predefined categories of queries: data can be queried using the traditional query languages (e.g. SQL or MDX). For these reasons the solutions proposed in the literature are generally not applicable to a coopetitive data warehouse application. Finally, as discussed in [15], *data quality* impacts the value of information. Therefore it is possible to alter the quality of provided data to reduce lv. In the case of coopetitive information system a possible strategy could be to reduce the quality of some data quality dimensions such as accuracy, timeliness or completeness to reduce lv. In the following section we discuss a methodology for designing a coopetitive data warehouse application, and subsequently a software architecture appropriate to satisfy the above mentioned critical high level requirements, as a result of the case study.

3 A Methodology

The methodology we introduced is an extension of the well known Kimball Lifecyle [23], and thus is a requirement-driven approach [24]. We start from this approach due to its diffusion in the development of real DWs. The second phase (*requirement elicitation*) is extended with regard to the original Kimball contribution. The next phase concerns *conceptual design*, producing the conceptual schema of the Operational Data Store(ODS) followed as parallel phases the *design of the Business Intelligence (BI) application*, *the global virtual view (GVV)* and traditional *design of DW and ETL*. The following phase is the deployment, which leads to the maintenance phase and/or to the development of a new data warehouse, in an incremental process. We now focus on the original contribution

related to the requirement elicitation phase, whereas the other phases of the methodology can be realized by means of existing techniques.

The purpose of the requirements elicitation is twofold. First, information requirements of the users of the DWA have to be determined using well known techniques, like goal-oriented approaches [25][26], or approaches based on use-cases [27]. Second, coopetitors have to form a board leading to the agreement of shared data. This task could last for some weeks, until an accepted compromise about the boundaries of the system is reached. In particular the agreement must include a definition of *what* information the system will extract, integrate and produce, and *how* it will do that, wrt the needs of preserving the privacy of such information.

The first task, namely the *concepts definition*, leads to a definition of the *what*, and ends with two outputs:

- a global model of the information assets to consider, at an abstract level. It represents the portion of global business to be analyzed, and can be then formalized using traditional visual languages such ER or UML.
- a definition of quality of shared and aggregate data to obtain a good balance between the need to reduce the loss value (see Section 2) and the usability of such data.

The goal of the second task, namely the *concepts classification*, which leads to a definition of the *how*, is to agree on the privacy requirements of each of the concepts considered in the global business model. The concepts are classified against a scale of security classes. We adopt the scale of five classes used in mandatory access control (MAC) systems: unclassified, classified, confidential, secret and top-secret. The clearance of each class is determined by the what type of intervention is applied to the information, with regard to each dimensions influencing lv (access control policy, privacy preserving strategies and quality of shared data). First coopetitors define possible values for each dimension (e.g. the allow or not to access data, values of quality dimensions to be considered), such values are projected on a coopetitive ipercube, then for each combination of the ipercube dimensions coopetitors defined the appropriate security classes to access to a specific concept by a specific class of users.

Starting from the coopetitive ipercube is then possible to classify shared data as follows:

1. *unclassified*: concept X can be shared freely.
2. *classified*: plain concept X can be shared with a intermediated quality level
3. *confidential*: concept X can be shared in an aggregate way with a intermediate quality level.
4. *secret*: concept X can be shared, in a way to preserve privacy, and with a low level quality.
5. *top secret*: concept X must not be accessed.

Results of these phases are used to design schema, ETL and implement access control policies.

In the above classification, unclassified concepts for members of a coopetitive organization could be the names of the product type, representing the unique set of information shared by all organizations. Concepts related to product type that are produced by only one of the organizations are confidential, the average price of selling evaluated over all clients of an organization is a secret information while the specific price of a product type applied to a specific client is (obviously) a top secret information. It is worth noting that the definition of a classification of information is the starting point to identify the part of data sources that each organization wants to share and, according to different level of clearances, enables the definition of the most appropriate policy for extracting, integrating and accessing such data.

Concerning the management of quality of exposed data we focused on timeliness of data. There is often a difference of time between the event (e.g. a sell of good or a purchase of raw materials) and the registration of its data into the information systems. For example a company can sell products daily, but the economic value of all sells is reported once a month through a unique fiscal document stored in the information system. Moreover data are modified due to several reasons including data quality errors; thus, there is the problem to identify time windows after which data is considered stable without any change. The definition of the most appropriate schedule for the data integration is also mandatory for building a good privacy enforcing system, and consequently to reduce the loss value described in Section 2. In fact, the value of information is related to its quality [15]. Thus, a degradation of the quality of data, that is an extended delay between the generation of a data in the local information systems and its registration into the integration system, reduces the value of the information exposed, thereby reducing the value lv, and consequently increasing the utility for an organization in sharing data. Notice that the timeliness is the most appropriate data quality dimension that can be exploited to preserve at same time the privacy and the utility of data over medium-long time period, that is the typical use of such data in traditional data warehouse application. Another reason to use of different timeliness for secret data is related to antitrust regulation: in particular, according to the Italian (and other countries) antitrust laws it is forbidden that competitive organizations share up-to-date information related to price of sell or buy because participants can apply price strategy or cartel agreements against the market.

4 The AOPUnoLombardia Case Study

We applied the above development process to the case of the AOP UnoLombardia that is an organization of fruit and vegetable producers. It is composed by 12 grower organizations (GO) including biggest company in the Italian and European market with brands like Bonduelle and Dimmidisi. AOP UnoLombardia represents about the 20% of the whole fruit and vegetables market in Italy. GOs of AOP UnoLombardia produce a wide range of vegetable including product of gamma I (fresh fruit and vegetables) and IV (vegetable cleaned and chopped

ready to eat). AOP UnoLombardia wanted to develop a coopetitive data warehouse application with the following goals:

- to obtain a unified analysis of selling of goods in term of price and amount of sold pieces
- to obtain a unified view of raw materials bought also due to the fact that farmers have often a important role in the organization

The GOs participating to the implementation of systems are five due to economic and time constraint defined by the project in which the application was developed. Anyway all GOs participate at the requirement elicitation phase. According to the methodology described in Section 3 during the requirement phase, we identified shared information and their level of clearance and timeliness. Shared information are related to the sell of product types and purchase of raw materials; in particular we focus on their volume and price starting from the beginning of July 2009. Product types represent the most common product sold by large scale retail such as salad, rocket, endive, spinach ready to eat. Possible users of the coopetitive data warehouse application are anonymous users, members of AOP UnoLombardia do not provide data, and active coopetitors (that is members providing data), Concerning the definition of the ipercube, we assume only two access control policies (allow/denied), privacy preserving techniques are mainly based on the aggregation of data by means of traditional mathematician formulas. The only considered quality dimension is the timeliness and possible values are delay of two weeks, one month and three months.
According to our methodology, during the requirement phase, coopetitors have agreed on the types of information to be shared, and defined their level of clearance by means of the security matrix shown in Table 2 (U = User type, P = Privacy, T = Timeliness). According to Table 2, for members of AOP UnoLombardia product types' names are *unclassified* and thus can be shared freely. *Confidential* data such as sales values and volumes can be shared in an aggregate form only. Sharing of *secret* information requires a reduction of quality, in addition to aggregation. Thus, for example, the records of purchased raw materials are extracted from the ODS three months after their registration into the data sources. Finally, *top-secret* data are too sensitive to be shared with members of AOP UnoLombardia.

In general all GOs were agreed that the name of their clients (i.e. specific the large scale retail) and the relationship between the clients and product sold are *top-secret* information. Concerning the sell of product types, such data are confidential data and they aggregated and show to different users according to their

Table 2. Security matrix for all concepts applied to members of AOP UnoLombardia

Clearance	*U*	*P*	*T*
Unclassified	Allow	Plain	Two weeks
Confidential	Allow	Aggregate	One month
Secret	Allow	Aggregate	Three months
Top secret	Denied	-	-

level of access. In our case potential users are coopetitiors, that is GO providing data, AOP UnoLombardia, that is all GO and anonymous users, including public administrations, journals and so on. Concerning raw materials, the price of purchase is considered secret information thus price data are available one month later for AOP UnoLombardia members providing data, three months later respect to the day of purchase for other AOP UnoLombardia members and they are not shown to anonymous users. Data related to the farmers selling raw material to GO are top-secret, thus personal data related to farmer company was keep anonymous directly by GO and any information related to them is not imported into the developed DW. The conceptual design phase produced the Entity-Relationship (ER) model [28] shown in Figure 1.

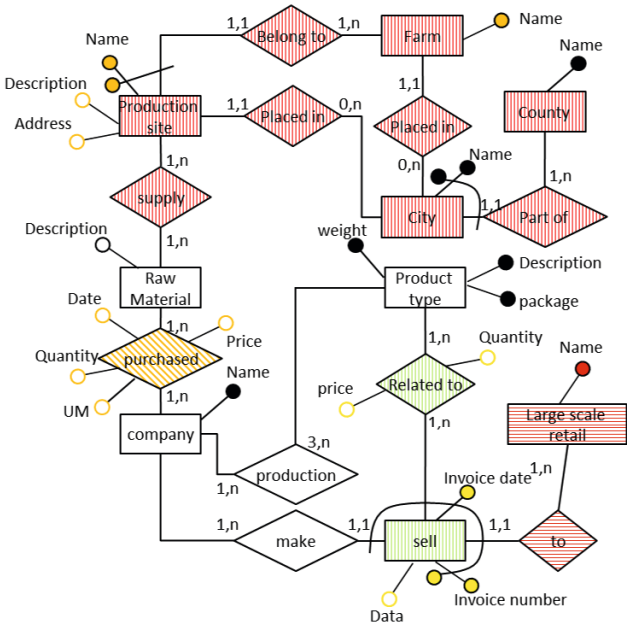

Fig. 1. Conceptual model

In Figure 1 unclassified data are show as white boxes, top secret information are shown by filling in involved entities and relationships with red vertical lines, confidential data ere those entities and relationships filled in by green horizontal lines and secret data are entities and relationships filled in by yellow slashed.
In the following Section we discuss and detail the resulting architecture of the coopetitive data warehouse applicationof AOP UnoLombardia.

4.1 Software Architecture

As for the development phase at AOP UnoLombardia, we argue that to achieve a maximization of *av* the system should not only integrate local data, but also must

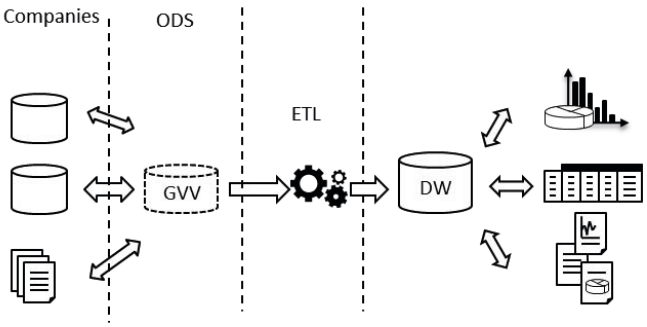

Fig. 2. CDW architecture

be able to exploit the integrated data in order to build new information useful for all coopetitors. For this to happen the system should provide suitable tools to flexibly and efficiently aggregate and analyse the integrated data over multiple dimensions. Given this goal, as said above, the choice to build a data warehouse application also considers the three above mentioned drivers that affect lv. In this section we discuss the architecture for the coopetitive data warehouse application, that is a data warehouse application architecture aimed at lowering lv by supporting *access control*, *privacy preservation* and *data quality*. The architecture results from the experience carried out at AOP UnoLombardiaas the most appropriate to satisfy both high level and context dependent requirements. We define the three levels software architecture shown in Figure 2. As for access control, the preservation of data ownership is the first issue considered in the definition of the coopetitive data warehouse application whose data coming from different organizations [29]. Indeed, coopetitors don't permit that sensitive local data are stored outside the organization boundaries. The solution chosen is to combine traditional Extraction, Transformation and Loading (ETL) techniques and Enterprise Information Integration (EII) techniques, with the goal of exploiting the advantages provided by both techniques. To reduce the loss of data ownership, we propose to use a EII system implementing a wrapper mediator architecture, and thus producing a virtual data integration [30]. Local data are still stored within the organization that produce and control them, and other organizations can only access to the integrated data by means of a virtual schema. The mediator component of the EII system provides a global virtual view (GVV) [31] that integrates the whole set of data sources. The mediator is then physically hosted by the organization of companies promoting the coopetitive data warehouse so that all companies have trust in it (otherwise it not makes sense to be part of it). Wrapper components allow the mediator to communicate with each data source, and to translate data models, query languages and dialects that are source-specific, into the ones used by the mediator. The coopetitive data warehouse application is feeded by an ETL component that extract data from the GVV. In particular the ETL is in charge of i) extracting integrated data from the GVV; ii) cleaning extracted data; iii) checking and monitoring their quality;

iv) transforming the structure of these data toward the dimensional model of coopetitive data warehouse application; v) loading the data into the coopetitive data warehouse application. As for privacy preservation, the design of coopetitive data warehouse application has to guarantee that sensitive information of a coopetitor are not revealed to other coopetitors or to outside users. It is important to note that architecture plays a fundamental role for the implementation of the proposed solutions, which is based on specific access control policies and manipulation of the quality of the data (e.g. reducing accuracy or timeliness [32]). Notice that according to our approach the GVV represents an Operational Data Store (ODS).

Virtual integration requires both the resolution of schema heterogeneities at design time and the resolution of instance conflicts at run time. Although this second issue usually makes virtual integration a hard task, in a coopetitive data warehouse application the complexity is reduced because of two peculiarities. First, the development is guided by an agreement between the organizations, which bounds the set of data they provide, both in terms of facts and values of the dimensions. Second, competing organizations are completely separated environments, with no overlapping among their business transactions. Thus, the local data provided by each organization are almost disjoint sets with regard to the facts, while they expose an high or complete overlapping with regard to dimension values. Thus, the fact tables are made up of horizontal partitions with each partition belonging to a single organization, while the agreement defines the design and the contents of the dimension tables. This means that instance level conflicts on the dimension tables are implicitly resolved at design time by the agreement, while the disjointness of the sets of data belonging to each participant guarantees the absence of instance level conflicts on the fact tables.

As a consequence, the integration of the fact tables comes down to a union of the facts extracted from the coopetitors' systems. For example suppose that the organizations agree on sharing the daily prices of the goods $p1$ and $p2$ and every organization will provide the same set of values with regard to the set of goods, namely the set $\{p1, p2\}$. Similarly, the sets of dates will be nearly identical. On the contrary, the sets of facts provided by each organization are disjoint, because each of them owns only its own sales. The mediator can leverage this particular feature of coopetitive data warehouse application to efficiently realize the virtual data integration, where efficiency problems at runtime are mostly due to the resolution of instance level conflicts. Indeed, on one hand, the fact that data sets representing facts provided by different organizations are disjoint allows to avoid any conflict. On the other hand, the integration of values of dimensions can't cause loss in efficiency because they are usually of several magnitude fewer than facts. In addition, if the values of a dimension are explicitly defined in the agreement (e.g. the dataset $\{p1, p2\}$ of the previous example) they don't need to be provided by the organizations and thus integrated. These values constitute a sort of *common vocabulary* for the given dimension and can be stored in an additional data source, supporting the integration operations. This data source includes also all the data structures needed to resolve instance level conflicts in

a declarative way. These structures, which we call *mapping tables*, maintain a one-to-many relationship between each *official* value in the common vocabulary and the corresponding representation in each data source.

The three levels architecture with ODS drops the latency between the time for data production and the time for its availability in the ODS: this side effect allows us to consider the ODS we propose as belonging to class I of the classification proposed by Inmon [33], thus being an enabling factor for defining *dashboards* able to show up-to-date gauges. Moreover it could be employed as a basis for *near real-time* coopetitive data warehouse application. As for this issue, it is worth noting that the traditional separation between OLTP and OLAP workloads is preserved because analytical processing is done only by the coopetitive data warehouse application, which is a separate physical system. Only the requests served directly by the ODS are in charge of the original data sources. These includes a) the queries needed by the ETL subsystem to extract the operational data, and b) sporadically analytical elaborations that involve up-to-date data. The coopetitive data warehouse application architecture guarantees also a good level of scalability in case of admission of new organizations in the coopetitive network. New data sources can be easily added to the GVV simply adding and/or changing the proper mappings.

Concerning dimensions affecting lv, the proposed architecture ensures that no organization can exploit the data provided by the coopetitive data warehouse application to infer single coopetitors' data. Indeed, for what concerns the access control policies, the proposed architecture foresees three different points of communication among architectural components. At the coopetitive data warehouse application level, users can access only a selected set of outputs (defined by the coopetitive data warehouse application application), and they cannot access directly to the operational data stage. We assume that users can be classified in different categories (e.g. coopetitors and external users) and they can access to different subset of outputs. Access control policy to ODS and local data can be implemented by restricting the access to computers hosting the mediator and the ETL respectively. So the access to a local data source is allowed only to the corresponding wrapper of the EII system. The communication between these two systems can be secured through a standard Virtual Private Network (VPN). The ETL component is the only software able to access to the ODS: such components must to be under the technical responsibility of the organization playing the role of third party among coopetitors. The use of secure communication channels and access policy control can guarantee the same security level of existing DBMS and DWA. For what concerns privacy preserving tasks, they are carried out by the ETL subsystem. Data of the coopetitors are aggregated to guarantee that the coopetitive data warehouse application contains only summary data belonging to the whole coopetitors' network, and not to single coopetitors. In this way let $n \geq 3$ the number of coopetitors, it is impossible for any single company to infer from aggregated data, specific data of any other coopetitors.

As for the quality of shared data it can be manipulated at different levels of the proposed architecture, for example if there is the need to reduce the timeliness

of shared data it is realized by the wrapper component, while if there is the need to reduce the accuracy of aggregated data it can be realized by the ETL component.

To implement the architecture we use the *open source* platform JBoss Teiid 7.3[2] (an EII paltform based on a wrapper mediator architecture). The mapping with the local source is realized within Teiid after the connection of the relational data stored in the information systems of the GO. To enforce the privacy of data the access to data source is realized over a secure channel. The implementation of the DW is realized by means of the open source platform Pentaho BI Suite Community Edition (CE) 3.7 [3]. Within the project we realized two different cubes: the first one related to the sell of references and the second one related to the purchase of raw material. We customize the Penthao interface and it allows users to make typical *roll-up*, *drill-down*, *drill-across*, *slice* and *dice* operations in automatic way.

5 Evaluation

The system is up and running by the end of May 2011, being used by both AOP UnoLombardia and all GO sharing data. The results of the integration of the data from 2009 are considered very important by AOP UnoLombardia members, providing them a real added value wrt the reduced possibility of data disclosure. They have now the possibility, with real data, to understand trends and make forecasts related to the prices they apply to big retail chains (clients) for the sales, and by the suppliers for the purchasing of raw materials.
As an example of it uses in real situation we report two analysis related to E.coli bacteria outbreak [4] registered at the end of May of 2011 in different countries of Europe (including German, France, but not Italy) where many people are died due to the E.coli bacteria found in some vegetables. News related to such outbreak produces a shock in the Italian public opinion that decided to not buy any fresh vegetables that can be in some case related to the E.coli bacteria (in such days several media declare that fresh vegetable could be affected by E.coli bacteria, even if it is not true). AOP UnoLombardia used the coopetitive data warehouse to evaluate:

- the reduction of the amount of product type sold to large-scale retail wrt the the previous year
- the reduction of purchase of raw material wrt the the previous year

Results of integrated data of AOP UnoLombardia are shown in figure 3 and 4 and they refers to the kind of salad called "Iceberg". By evaluating results of Figure 3 it is worth noting that there is a big difference between the amount of sold product related to iceberg salad in June and July 2010 (at the left of the

[2] http://www.jboss.org/teiid
[3] http://community.pentaho.com/
[4] http://en.wikipedia.org/wiki/2011_E._coli_O104:H4_outbreak

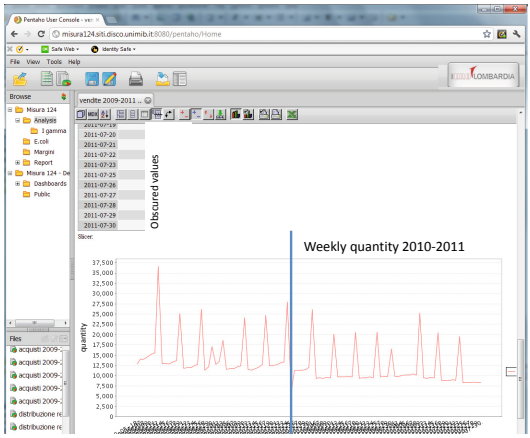

Fig. 3. Trends in selling of Iceberg salad ready to eat

blue vertical line) and the amount of same product type sold in the same two months of 2011. The reduction is about 29%, in some weeks the number of sold product was reduced by a 38% wrt the same period of the previous year. Notice that only in September 2011 the amount of sold items had the same value of the previous year. Notice that it is possibile to see only a slightly reduction of raw Iceberg purchased in the same period the amount of purchased salad (see the left and the right side of the blue line in figure 4); this is due to the fact that GOalready signed contracts with farmers to buy almost predefined amount of raw material and it is not easy to change such contracts (due to economic penalties). This represented a huge problem for AOP UnoLombardia members due to overload of raw material that it is not sold to clients. In such a way AOP UnoLombardia was able to give a precise value of the economic loss related to

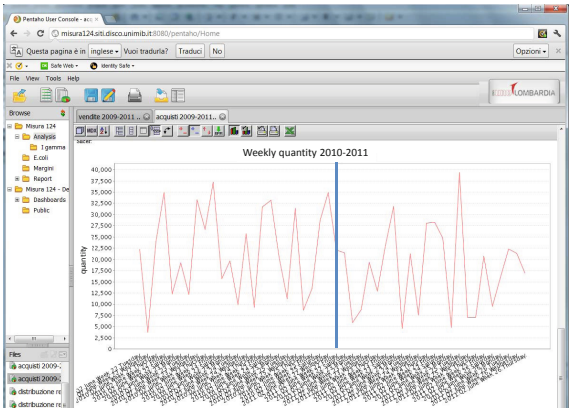

Fig. 4. Trends in purchase of raw materials related to Iceberg salad

the E.coli outbreak, that was impossible to obtain without the developed system with the same level of precision. It is worth noting that in this case data are real (under the explicit authorization of AOP UnoLombardia).

6 Related Works and Conclusions

Even if the literature on both data warehouse and privacy preserving data mining is very large [34], to the best of our knowledge this is the first work related to the definition of a data warehouse application in a coopetitive environment. Most of the existing problems defined in the literature considering a single relational schema and no proposal was applied to a real case. Existing techniques based on cryptography are too expensive to be applied in real context[35]. In this paper we have discussed a real implementation of a data warehouse application in a coopetitive environment. First, we modeled the coopetition by means of game theory, subsequently we defined a methodological framework and, and finally we shown through a case study how a coopetitive data warehouse application as integrated information solution is able to reduce the loss related to the disclosure of information. Future work includes the extension of the experience to all GOs of AOP UnoLombardia and the evaluation of privacy preserving techniques applied to statistical database, even if as reported in [36] there is no universal solution due to, among others, 1) it is difficult to determine the a priori knowledge of a malicious user, 2) users may collude, and 3) there is a computationally challenging problem to provide good trade off between data privacy and data utility.

Acknowledgments. This paper was partially founded by the Region Lombardia Project "Advanced primary" within the Measure 124 found. Authors are very grateful to AOP UnoLombardia for the support and the authorization to show part of their data.

References

1. Brandenburger, A.M., Nalebuff, B.: Co-opetition. Doubleday and Company, New York (1996)
2. Gnyawali, D.R., Madhavan, R.: Cooperative Networks and Competitive Dynamics: A Structural Embeddedness Perspective. The Academy of Management Review 26(3), 431–445 (2001)
3. Levy, M., Loebecke, Claudiaand Powell, P.: Smes, co-opetition and knowledge sharing: the role of information systems. European Journal of Information Systems 12(1), 14–25 (2003)
4. Ghobadi, S., DAmbra, J.: Coopetitive knowledge sharing: An analytical review of literature. The Electronic Journal of Knowledge Management 09(4), 307–317 (2011)
5. Thoo, E., Friedman, T.: The Logical Data Warehouse will be a Key Scenario for using Data Federation (2012)

6. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly 28(1), 75–105 (2004)
7. Iivari, J., Venable, J.: Action research and design science research - seemingly similar but decisively dissimilar. In: ECIS, pp. 1642–1653 (2009)
8. Fudenberg, D., Tirole, J.: Game Theory. MIT Press (1991)
9. Nash, J.: Equilibrium points in n-person games. Proceedings of the National Academy of Sciences of the United States of America, 48–49 (1950)
10. Bassan, B., Gossner, O., Scarsini, M., Zamir, S.: Positive value of information in games. International Journal of Game Theory 32, 17–31 (2003)
11. Schlee, E.: The value of information in anticipated utility theory. Journal of Risk and Uncertainty 3, 83–92 (1990)
12. Schlee, E.: The value of perfect information in nonlinear utility theory. Theory and Decision 30, 127–131 (1991)
13. Lehrer, E., Rosenberg, D.: What restrictions do bayesian games impose on the value of information? Journal of Mathematical Economics 42(3), 343–357 (2006)
14. Ahituv, N.: A systematic approach toward assessing the value of an information system. MIS Quarterly 4, 61–75 (1980)
15. Batini, C., Cappiello, C., Francalanci, C., Maurino, A., Viscusi, G.: A capacity and value based model for data architectures adopting integration technologies. In: AMCIS, p. 237 (2011)
16. Ahituv, N.: Assessing the value of information. In: ICIS, pp. 315–325 (1989)
17. Moody, D.L., Walsh, P.: Measuring the value of information - an asset valuation approach. In: ECIS, pp. 496–512 (1999)
18. Ahituv, N., Greenstein, G.: The impact of accessibility on the value of information and the productivity paradox. European Journal of Operational Research 161(2), 505–524 (2005)
19. Tomkins, C.: Interdependencies, trust and information in relationships, alliances and networks. Accounting, Organizations and Society 26(2), 161–191 (2001)
20. Kai-Lung, H., Hock Hai, T., Sang-Yong Tom, L.: The value of privacy assurance: an exploratory field experiment. MIS Q. 31(1), 19–33 (2007)
21. Pavlou, P.A.: State of the information privacy literature: Where are we now and where should we go? MIS Quarterly 35, 977–988 (2011)
22. Aggarwal, C.C., Yu, P.S., et al.: Privacy-Preserving Data Mining. Springer (2008)
23. Kimbal, R., et al.: The DW Lifecycle Toolkit (2008)
24. Winter, R., Strauch, B.: A method for demand-driven information requirements analysis in data warehousing projects. In: Proc. HICSS, pp. 1359–1365 (2003)
25. Prakash, N., Gosain, A.: Requirements driven data warehouse development. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681, pp. 13–17. Springer, Heidelberg (2003)
26. Giorgini, P., Rizzi, S., Garzetti, M.: Goal-oriented requirement analysis for data warehouse design. In: Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP, DOLAP 2005, pp. 47–56 (2005)
27. Bruckner, R.M., List, B., Schiefer, J.: Developing requirements for data warehouse systems with use cases. In: Proceedings of the 7th Americas Conference on Information Systems, pp. 329–335 (2001)
28. Batini, C., Ceri, S., Navathe, S.: Conceptual database design: an Entity-relationship approach. Benjamin-Cummings Publishing Co., Inc., Redwood City (1991)
29. Alstyne, M.V., Brynjolfsson, E., Madnick, S.: Why not one big database? principles for data ownership. Decision Support Systems 15(4), 267–284 (1995)

30. Wiederhold, G.: Mediators in the architecture of future information systems. Computer 25(3), 38–49 (1992)
31. Lenzerini, M.: Data integration: A theoretical perspective. In: PODS, pp. 233–246 (2002)
32. Batini, C., Cappiello, C., Francalanci, C., Maurino, A.: Methodologies for data quality assessment and improvement. ACM Comput. Surv. 41, 16:1–16:52 (2009)
33. Inmon, B.: Building the Data Warehouse, 4th edn. John Wiley & Sons, Ltd. (2005)
34. Agrawal, R., Srikant, R.: Privacy-preserving data mining. SIGMOD Rec. 29, 439–450 (2000)
35. Atallah, M.J., Frikken, K.B.: Securely outsourcing linear algebra computations. In: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2010, pp. 48–59. ACM, New York (2010)
36. Adam, N.R., Lu, H., Vaidya, J., Shafiq, B.: Statistical databases. In: van Tilborg, H.C.A., Jajodia, S. (eds.) Encyclopedia of Cryptography and Security, 2nd edn., pp. 1256–1260. Springer (2011)