9

Collaborative Business Intelligence

Stefano Rizzi

Department of Electronics, Computer Sciences and Systems (DEIS)
University of Bologna
Bologna, Italy
stefano.rizzi@unibo.it

Summary. The idea of collaborative BI is to extend the decision-making process beyond the company boundaries thanks to cooperation and data sharing with other companies and organizations. Unfortunately, traditional BI applications are aimed at serving individual companies, and they cannot operate over networks of companies characterized by an organizational, lexical, and semantic heterogeneity. In such distributed business scenarios, to maximize the effectiveness of monitoring and decision making processes there is a need for innovative approaches and architectures. Data warehouse integration is an enabling technique for collaborative BI, and has been investigated along three main directions: warehousing approaches, where the integrated data are physically materialized, federative approaches, where the integration is virtual and based on a global schema, and peer-to-peer approaches, that do not rely on a global schema to integrate the component data warehouses. In this paper we explore and compare these three directions by surveying the available work in the literature. Then we outline a new peer-to-peer framework, called Business Intelligence Network, where peers expose querying functionalities aimed at sharing business information for the decision-making process. The main features of this framework are decentralization, scalability, and full autonomy of peers.

Keywords: business intelligence, distributed databases, query reformulation, peer-to-peer architectures.

9.1 Introduction

A new generation of business intelligence (BI) systems has been emerging during the last few years to meet the new, sophisticated requirements of business users. The term *BI 2.0* has been coined to denote these systems; among their characterizing trends, we mention:

- *BI as a service*, where BI applications are hosted as a service provided to business users across the Internet.
- *Real-time BI*, where information about business operations is delivered as they occur, with near-0 latency.
- *Situational BI*, where information in an enterprise data warehouse is completed and enriched by correlating it with external information that may

M.-A. Aufaure and E. Zimányi (Eds.): eBISS 2011, LNBIP 96, pp. 186–205, 2012.

come from the corporate intranet, be acquired from some external vendor, or be derived from the internet.
- *Pervasive BI*, where information can be easily and timely accessed through devices with different computation and visualization capabilities, and with sophisticated and customizable presentations, by everyone in the organization.
- *Collaborative BI*, where a company information assets are empowered thanks to cooperation and data sharing with other companies and organizations, so that the decision-making process is extended beyond the company boundaries.

In particular, collaborative BI has been predicted to be the main BI trend for 2011 [1]. From the Wikipedia:

"Collaboration is working together to achieve a goal *[...]. It is a recursive process where two or more people or organizations work together to realize shared goals —this is more than the intersection of common goals, but a deep, collective, determination to reach an identical objective—* by sharing knowledge*, learning and building consensus. [...] Teams that work collaboratively can obtain* greater resources, recognition and reward *when facing competition for finite resources."*

Indeed, cooperation is seen by companies as one of the major means for increasing flexibility, competitiveness, and efficiency so as to survive in today uncertain and changing market. Companies need strategic information about the outer world, for instance about trading partners and related business areas [2]. Users need to access information anywhere it can be found, by locating it through a semantic process and performing integration on the fly. This is particularly relevant in inter-business collaborative contexts where companies organize and coordinate themselves to share opportunities, respecting their own autonomy and heterogeneity but pursuing a common goal.

Unfortunately, most information systems were devised for individual companies and for operating on internal information, and they give limited support to inter-company cooperation. In the same way, traditional BI applications are aimed at serving individual companies, and they cannot operate over networks of companies characterized by an organizational, lexical, and semantic heterogeneity. In such a complex and distributed business scenario, to maximize the effectiveness of monitoring and decision making processes there is a need for innovative approaches and architectures.

Data warehouse integration is an enabling technique for collaborative BI. It provides a broader base for decision-support and knowledge discovery than each single data warehouse could offer. Large corporations integrate their separately-developed departmental data warehouses; newly merged companies integrate their data warehouses into a central data warehouse;

autonomous but related organizations join together their data warehouses to enforce the decision making process [3].

Although the integration of heterogeneous databases has been widely discussed in the literature, only a few works are specifically focused on strategies for data warehouse integration. Two categories of approaches were mainly devised: *warehousing* approaches, where the integrated data are physically materialized, and *federative* approaches, where integration is virtual. In both cases, it is assumed that all components to be integrated share the same schema, or at least that a global schema is given. This assumption is perfectly reasonable in business contexts where a common view of the business is shared, or where one of the component parties has a clear leadership. In contexts where the different parties have a common interest in collaborating while fully preserving their autonomy and their view of business, defining a global schema is often unfeasible. To cope with this, the category of *peer-to-peer* (P2P) approaches, that do not rely on a global schema to integrate the component data warehouses, has been emerging during the last few years. In P2P approaches, each peer can formulate queries also involving the other peers, typically based on a set of mappings that establish semantic relationships between the peers' schemata.

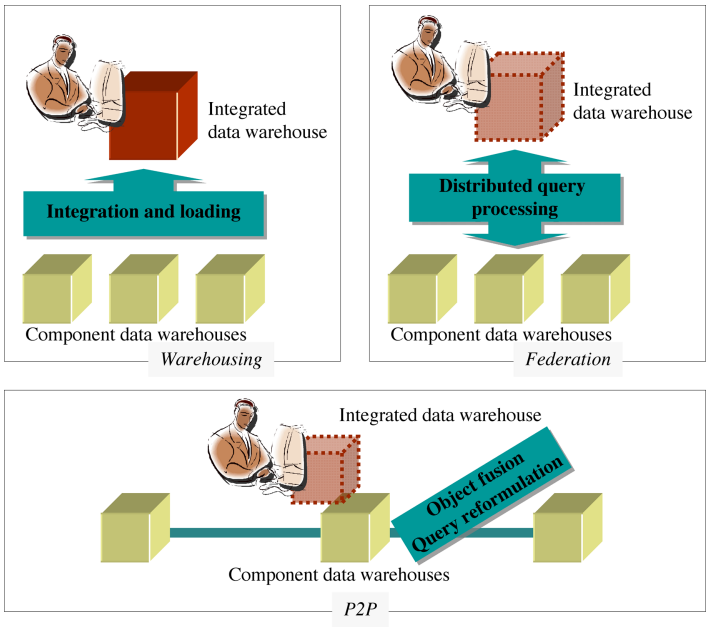

Fig. 9.1. Three approaches to collaborative BI

In this paper we compare these three categories by exploring the available works in the literature. In particular, after surveying the related literature

in the OLTP field in Section 9.2, in Sections 9.3, 9.4, and 9.5 we survey the warehousing, federative, and P2P approaches, respectively. Then in Section 9.6 we outline a new peer-to-peer framework, called Business Intelligence Network, where peers expose querying functionalities aimed at sharing business information for the decision-making process. The main features of this framework are decentralization, scalability, and full autonomy of peers. Finally, in Section 9.7 the conclusions are drawn.

9.2 Related Literature for OLTP

In the OLTP context, the research area sharing most similarities with warehousing approaches to collaborative BI is *data exchange*. In data exchange, data structured under one source schema must be restructured and translated into an instance of a different target schema, that is materialized [4]. In this scenario, the target schema is often independently created and comes with its own constraints that have to be satisfied.

On the other hand, federative approaches have their OLTP counterpart in *data integration systems*. Data from different sources are combined to give users a unified view [5]; in this way, users are freed from having to locate individual sources, learn their specific interaction details, and manually combine the data [6]. The unified view that reconciles the sources is represented by a global schema. In this case query processing requires a reformulation step: a query over the global, target schema has to be reformulated in terms of a set of queries over the sources.

Finally, P2P approaches to collaborative BI are related to the decentralized sharing of OLTP data between autonomous sources, that has been deeply studied in the context of *Peer Data Management Systems* (PDMSs). PDMSs were born as an evolution of mediator systems in the data integration field [7] and generalize data exchange settings [8]. A PDMS consists of a set of peers, each with an associated schema representing its domain of interest; peer mediation is implemented by means of semantic mappings between portions of schemata that are local to a pair or a small set of peers. Every peer can act freely on its data, and also access data stored by other peers without having to learn their schema and even without a mediated schema [9]. In a PDMS there is no a priori distinction between source and target, since a peer may simultaneously act as a distributor of data (thus, a source peer) and a recipient of data (thus, a target peer). As in the case of data integration systems, in a PDMS data remain at the sources and queries processing entails query reformulation over the peer schemata.

In all these contexts, modeling the relationships (*mappings*) between source and target schemata is a crucial aspect. Research in the data integration area has provided rich and well-understood schema mediation languages [5] to this end. The two commonly used formalisms are the *global-as-view* (GAV) approach, in which the mediated (global) schema is defined as a set of views over the data sources, and the *local-as-view* (LAV) approach, in which the

contents of data sources are described as views over the mediated schema. Depending on the kind of formalism adopted, GAV or LAV, queries posed to the system are answered differently, namely by means of query unfolding or query rewriting techniques [10], respectively. In a data exchange setting, assertions between a source query and a target query are used to specify what source data should appear in the target and how. These assertions can be represented neither in the LAV nor in the GAV formalisms, but rather they can be thought of as GLAV (*global-and-local-as-view*) [4]. A structural characterization of schema mapping languages is provided in [11], together with a list of the basic tasks that all languages ought to support.

In distributed OLTP environments, the schema mapping generation phase and the preceding schema matching phase pose new issues with reference to simpler centralized contexts: consistency problems are studied in [12] and innovative learning techniques are presented in [13]. Other studies in the field have focused on integrating the computation of core solutions in the mapping generation process, aimed at determining redundancy-free mappings in data exchange settings [14, 15].

Declaring useful mappings in the OLAP context necessarily requires also the level of instances to be taken into account. Unfortunately, in the OLTP literature the definition of mappings is typically done at the schema level, and the problem of managing differences in data formats has only marginally been considered. A seminal paper regarding this topic is [16], where constraint queries are translated across heterogeneous information sources taking into account differences in operators and data formats.

A related problem is that of reconciliation of results, that takes a primary role in federative and P2P approaches. In the OLTP context this issue is referred to as *object fusion* [17]. This involves grouping together information (from the same or different sources) about the same real-world entity. In doing this fusion, the mediator may also "refine" the information by removing redundancies, resolving inconsistencies between sources in favor of the most reliable source, and so on.

9.3 Warehousing Approaches

As already mentioned, in this family of approaches the data that result from the process of integrating a set of component data warehouses according to a global schema are materialized. The main drawback of these approaches is that they can hardly support dynamic scenarios like those of mergers and acquisitions.

An approach in this direction is the one proposed in [18]. Given two dimensions belonging to different data marts where a set of mappings between corresponding levels has been manually declared or automatically inferred, three properties (namely *coherence*, *soundness*, and *consistency*) that enable a compatibility check between the two dimensions are defined. A technique

that combines the contents of the dimensions to be integrated is then used to derive a materialized view that includes the component data marts.

A hybrid approach between the warehouse and the federation approach is suggested in [19] as a way to obtain a more flexible and applicable architecture. The idea is to aggregate selected data from the component data warehouses as materialized views and cache them at a federation server to improve query performance; a set of *materialized query tables* are recommended for the benefits of load distribution and easy maintenance of aggregated data.

Another borderline approach is proposed in [20]: while fact data are not physically integrated, a central *dimension repository* is used to replicate dimensional data (according to a global schema) from the component data warehouses, aimed at increasing querying efficiency. To effectively cope with evolutions in the schema of the components, a fact algebra and a dimension algebra are used in this approach for declaring maintainable mappings between the component schemata.

9.4 Federative Approaches

A *federated data warehouse*, sometimes also called *distributed data warehouse*, is a logical integration of data warehouses that provides transparent access to the component data warehouses across the different functions of an organization. This is achieved through a global schema that represents the common business model of the organization [21]. Differently from warehousing approaches, the integrated data are not physically stored, so queries formulated on the global schema must be rewritten on the component schemata. This adds complexity to the query management framework, but enables more flexible architectures where new component data warehouses can be dynamically inserted.

A distributed data warehouse architecture is outlined in [22], and a prototype named CubeStar for distributed processing of OLAP queries is introduced. CubeStar includes a middleware layer in charge of making the details of data distribution transparent to the front-end layer, by generating optimized distributed execution plans for user queries.

A distributed data warehouse architecture is considered also in [23] as a solution for contexts where the inherently distributed nature of the data collection process and the huge amount of data extracted make the adoption of a central repository impractical. The Skalla system for distributed query processing is proposed, with particular emphasis on techniques for optimizing both local processing and communication costs; however, since it is assumed that all collection points share the same schema, the approach cannot be used to cope with heterogeneous settings.

In the context of a federated architecture, with specific reference to the healthcare domain, the work in [24, 3] presents an algorithm for matching heterogeneous multidimensional structures, possibly characterized by different granularities for data. Mappings between the local schemata of the data

warehouses to be integrated and a given global schema are discovered in a semi-automated manner, based on a measure of similarity between complex concepts.

A process to build an integrated view of a set of data warehouses is outlined in [25]. This integrated view is defined as the largest common schema to all the components, and its instances are obtained by merging the instances of the components.

In [18], the problem of virtual integration of heterogeneous data marts is faced in a loosely-coupled scenario where there is a need for identifying the common information (intuitively, the intersection) between the components while preserving their autonomy. A set of rules to check for dimension compatibility are declared first, then drill-across queries are used to correlate on-the-fly the component data marts.

A *multi data warehouse* system is introduced in [26] as one relying on a distributed architecture where users are enabled to directly access the heterogeneous schemata of the component data warehouses, which makes the coupling between the components looser than in federated data warehouses. A SQL-MDi query language is proposed to transform a cube in order to make it compatible with a global, virtual cube and ready for integration. Specific attention is devoted to solving schema and instance conflicts among the different components.

An XML-based framework for supporting interoperability of heterogeneous data warehouses in a federation scenario is described in [27]. In the proposed architecture, a *federated layer* allows for restructuring and merging local data and schemas to provide a global, single view of the component data warehouses to the end users. XML is used both to represent the local data warehouse schemata, the global schema, and the mapping between them.

Another XML-based approach is the one in [28], that discusses the possible conflicts arising when heterogeneous data warehouses are integrated and proposes solutions to resolve the semantic discrepancies. Data cubes are transformed into XML documents and queried under a global view.

XML *topic maps* are used in [29] to integrate the information stored in distributed data warehouses. The schema integration process is based on merging local topic maps to generate global topic maps, taking different types of semantic conflicts into account.

A different approach is presented in [30], that introduces an architecture for *hierarchically distributed data warehouses* where component data warehouses are organized into a tree and data are progressively summarized level over level. A local OLAP query can be posed at any node of the tree, it is rewritten on remote nodes, and the results are merged.

9.5 Peer-to-Peer Approaches

Though federative approaches support more flexible and dynamic architectures than warehousing ones, still they do not fully preserve the autonomy

of individual actors. In complex business scenarios where no leadership can be established among a set of actors interested in cooperating, to maximize the effectiveness of monitoring and decision making processes there is a need for truly decentralized approaches. This can be achieved by relying on P2P architectures.

In [31, 32], the authors introduced the idea of using a P2P architecture for warehousing XML content. In their view, a P2P warehouse is not different from a centralized one from the logical point of view, while from the physical point of view information is distributed over a set of heterogeneous and autonomous peers rather than centralized. Because of this, query processing necessarily requires distributed computation. Among the advantages of this approach, we mention ownership (each peer has full control over its information) and dynamicity (peers can transparently enter and leave the system). How to map the local schema of each peer onto each other is one of the open problems.

The approach proposed in [33] reformulates XML queries over a set of peers hosting XML databases with heterogeneous (and possibly conflicting) schemata, in the absence of a global schema. Reformulation is based on mapping rules inferred from informal schema correspondences.

In [34, 35] the authors present a model for multidimensional data distributed across a P2P network, together with a mapping-based technique for rewriting OLAP queries over peers. In presence of conflicting dimension members, an approach based on belief revision is proposed to revise the instance of the source peer's dimension and adapt it to the instance of the target peer's dimension.

Another work centered on interoperability issues among heterogeneous data warehouses is the one by [36], that emphasizes the importance of a semantic layer to enable communication among different components. This approach supports the exchange of business calculation definitions and allows for their automatic linking to specific component data warehouses through semantic reasoning. Three models are suggested: a business ontology, a data warehouse ontology, and a mapping ontology between them.

As to performance aspects, in [37] the authors propose a P2P architecture for supporting OLAP queries focusing on the definition of a caching model to make the query rewriting process more efficient. They also define adaptive techniques that dynamically reconfigure the network structure in order to minimize the query cost.

Finally, as to the data reconciliation, a typical requirement in collaborative BI is the merging of results at different levels of aggregation. In this direction, the work proposed in [38] discusses a general approach on the use of aggregation operations in information fusion processes and suggests practical rules to be applied in common scenarios.

9.6 Business Intelligence Networks

In this section we describe a new framework to collaborative BI, called *Business Intelligence Network* (BIN), based on a P2P architecture [39]. BINs enable BI functionalities to be shared over networks of companies that, though they may operate in different geographical and business contexts, are chasing mutual advantages by acting in a conscious and agreed upon way. A BIN is based on a network of peers, one for each company participating in the consortium; peers are equipped with independent BI platforms that expose some functionalities aimed at sharing business information for the decision-making process, in order to create new knowledge (Figure 9.2). Remarkably, since each peer is allowed to define and change the set of shared information as well as its own terminology and schema without being subject to a shared schema, the BIN approach fully preserves peer autonomy. Besides, the BIN architecture is completely decentralized and scalable to cope with business contexts where the number of participants, the complexity of business models, and the user workload are unknown a priori and may change in time.

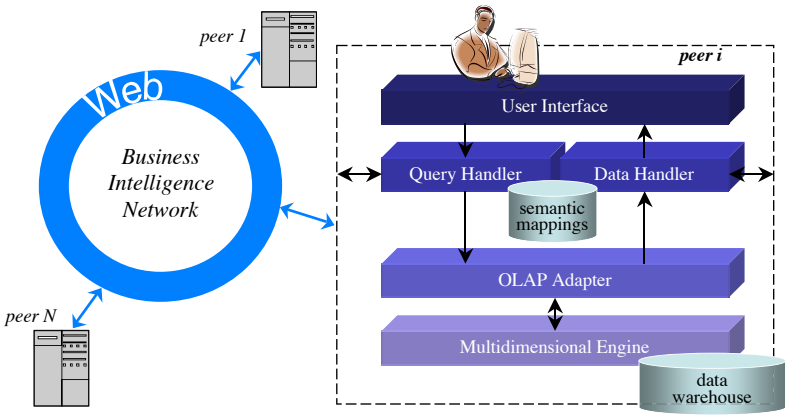

Fig. 9.2. A BIN architecture

The main benefits the BIN approach aims at delivering to the corporate world are (1) the possibility of building new inter-organizational relationships and coordination approaches, and (2) the ability to efficiently manage inter-company processes and safely share management information. Other practical benefits arising from activating a BIN depend on the specific corporate context; for instance, in companies that belong to the same supply-chain or operate in the same market, the (partial) business information sharing is required by users to allow inter-company processes to be monitored and markets to be accurately controlled.

The core idea of a BIN is that of enabling users to transparently access business information distributed over the network. A typical interaction sequence is the following:

1. A user formulates an OLAP query q by accessing the local multidimensional schema exposed by her peer, p.
2. Query q is processed locally on the data warehouse of p.
3. At the same time q is forwarded to the network.
4. Each involved peer locally processes the query on its data warehouse and returns its results to p.
5. The results are integrated and returned to the user.

The local multidimensional schemata of peers are typically heterogeneous. So, during distributed query processing, before a query issued on a peer can be forwarded to the network it must be first *reformulated* according to the multidimensional schemata of the source peers. Data are then extracted from each source peer and are mapped onto the schema of the querying peer, that plays the role of the target.

In line with the approach adopted in *Peer Data Management Systems* (PDMSs) [7], query reformulation in a BIN is based on *semantic mappings* that mediate between the different multidimensional schemata exposed by two peers, i.e., they describe how the concepts in the multidimensional schema of the target peer map onto those of the source peer. Direct mappings cannot be realistically defined for all the possible couples of peers. So, to enhance information sharing, a query q issued on p is forwarded to the network by first sending it to (a subset of) the immediate neighbors of p, then to their immediate neighbors, and so on. In this way, q undergoes a chain of reformulations along the peers it reaches, and results are collected from any peer that is connected to p through a path of semantic mappings. This process is sketched in Figure 9.3.

The approach outlined above is reflected by the internal architecture of each peer, sketched in the right side of Figure 9.2, whose components are:

1. *User Interface.* A web-based component that manages bidirectional interaction with users, who use it to visually formulate OLAP queries on the local multidimensional schema and explore query results.
2. *Query Handler.* This component receives an OLAP query from either the user interface or a neighboring peer on the network, sends that query to the OLAP adapter to have it locally answered, reformulates it onto the neighboring peers (using the available semantic mappings), and transmits it to those peers.
3. *Data Handler.* When the peer is processing a query that was locally formulated (i.e., it is acting as a target peer), the data handler collects query results from the OLAP adapter and from the source peers, integrates them, and returns them to the user interface. When the peer is processing a query that was formulated on some other peer p (i.e., it is

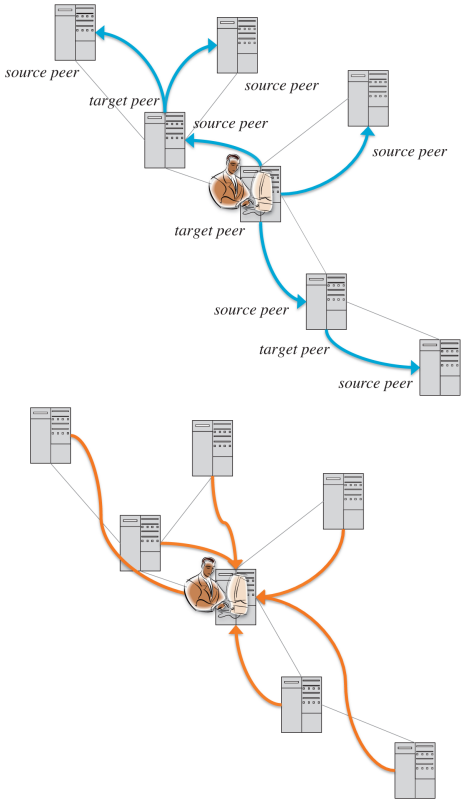

Fig. 9.3. Recursive query reformulation (left) and collection of results (right)

acting as a source peer), the data handler just collects local query results from the OLAP adapter and returns them to the target peer p.

4. *OLAP Adapter*. This component adapts queries received from the query handler to the querying interface exposed by the local multidimensional engine.
5. *Multidimensional Engine*. It manages the local data warehouse according to the multidimensional schema representing the peer's view of the business, and provides MDX-like query answering functionalities.

Interactions between peers are based on a message-passing protocol.

9.6.1 Mapping Language

Reformulation of OLAP queries first of all requires a language for properly expressing the *semantic mappings* between each couple of neighboring peers;

this language must accommodate the peculiar characteristics of the multidimensional model, on which the representation of business information at each peer is founded. The basic requirements for the mapping language are:

1. The asymmetry between dimensions and measures should be reflected in the mapping language by providing different predicates for mapping dimensions/attributes and measures.
2. Since aggregation is inherently part of the multidimensional model, the mapping language should be capable of specifying the relationship between two attributes of different multidimensional schemata in terms of their granularity.
3. A measure is inherently associated with an aggregation operator. Then, when mapping a measure onto another, their aggregation operators must be taken into account to avoid the risk of inconsistent query reformulations.
4. Declaring useful mappings in the BI context necessarily requires also the instance level to be taken into account. This can be done if there is a known way to transcode values of an attribute/measure belonging to a multidimensional schema into values of an attribute/measure belonging to another multidimensional schema.

The language used in a BIN to express how the multidimensional schema $\mathcal{M}_s$ of a source peer s maps onto the multidimensional schema $\mathcal{M}_t$ of a target peer t includes five *mapping predicates*, that will be explained below. In general, a mapping establishes a semantic relationship from one or more concepts (either measures or attributes) of $\mathcal{M}_s$ to one or more concepts of $\mathcal{M}_t$, and enables a BIN query formulated on $\mathcal{M}_t$ to be reformulated on $\mathcal{M}_s$. Optionally, a mapping involving attributes can be annotated with a *transcoding function* that specifies how values of the target concepts can be obtained from values of the source concepts. If this function is available, it is used to increase the reformulation effectiveness.

- `same` predicate: μ_t `same`$_{expr}$ M_s, where $\mu_t = \langle m_t, \alpha_t \rangle$ is a metric[1] of $\mathcal{M}_t$, M_s is a subset of measures of $\mathcal{M}_s$, and *expr* is an expression involving the measures in M_s. This mapping predicate is used to state that whenever m_t is asked in a query on $\mathcal{M}_t$ using α_t, it can be rewritten as *expr* on $\mathcal{M}_s$.
- `equi-level` predicate: P_t `equi-level`$_f$ P_s, where P_t and P_s are sets of attributes of $\mathcal{M}_t$ and $\mathcal{M}_s$, respectively. This predicate is used to state that P_t has the same semantics and granularity as P_s. Optionally, it can be annotated with an injective transcoding $f : Dom(P_s) \rightarrow Dom(P_t)$ that establishes a one-to-one relation between tuples of values of P_s and P_t, and is used to integrate data returned by the source and target peers.

[1] A *metric* of a multidimensional schema $\mathcal{M}$ is a couple $\mu = \langle m, \alpha \rangle$, where m is a measure and α is a valid aggregation operator for m.

- `roll-up` predicate: P_t `roll-up`$_f$ P_s. This predicate states that P_t is a roll-up of (i.e., it aggregates) P_s. Optionally, it can be annotated with a non-injective transcoding $f : Dom(P_s) \rightarrow Dom(P_t)$ that establishes a many-to-one relation between tuples of values of P_s and P_t, and is used to aggregate data returned by the source peer and integrate them with data returned by the target peer.
- `drill-down` predicate: P_t `drill-down`$_f$ P_s. This predicate is used to state that P_t is a drill-down of (i.e., it disaggregates) P_s. Optionally, it can be annotated with a non-injective transcoding $f : Dom(P_t) \rightarrow Dom(P_s)$ that establishes a one-to-many relation between tuples of values of P_s and P_t. The transcoding f cannot be used to integrate data returned by t and s because this would require disaggregating data returned by s, which obviously cannot be done univocally; however, it can be used to reformulate the selection predicates expressed at t onto s.
- `related` predicate: P_t `related` P_s. This predicate is used to state that P_t and P_s tuples of values have a many-to-many relationship.

Example 1. As a working example, we consider a BIN for sharing information about funded research projects among European nations. Figure 9.4 shows the multidimensional schemata of related facts at the peers in London and Rome, using the Dimensional Fact Model notation [40]; small circles represent attributes, while measures are listed inside the fact boxes. Note that, while an event in the Rome schema corresponds to the funding given to each research unit within a project, in the London schema an event aggregates the projects by their coordinator. Figure 9.4 also shows some of the mappings that can be

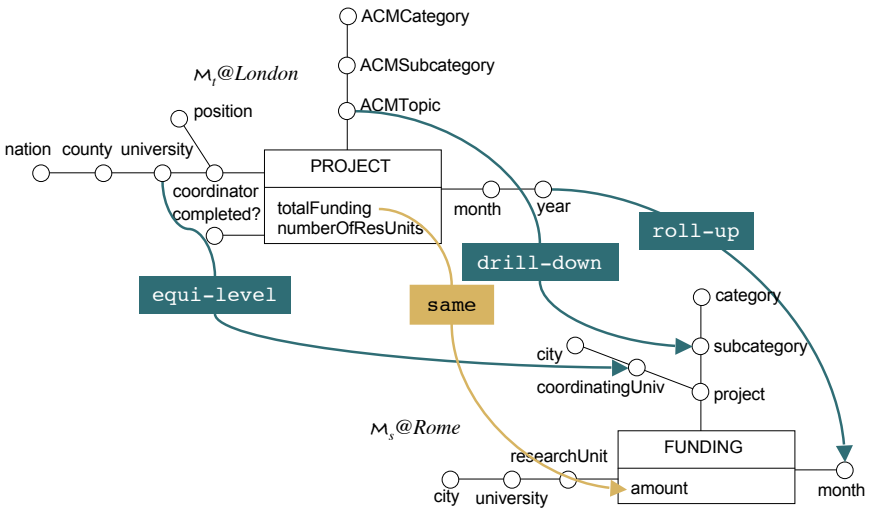

Fig. 9.4. Multidimensional schemata of related facts at two peers

defined to reformulate queries expressed in London (target peer) according to the schema adopted in Rome (source peer). As examples of transcodings, consider the function that associates each topic in the ACM classification with a subcategory and the one that associates each month with its year, used to annotate mappings ACMTopic `drill-down` subcategory and year `roll-up` month, respectively. Similarly, the `same` mapping between totalFunding and amount is annotated with an expression that converts euros into pounds.

9.6.2 Query Reformulation

Reformulation takes as input an OLAP query on a target schema $\mathcal{M}_t$ as well as the mappings between $\mathcal{M}_t$ and the schema of one of its neighbor peers, the source schema $\mathcal{M}_s$, to output an OLAP query that refers only to $\mathcal{M}_s$. The reformulation framework we propose is based on a relational setting, as depicted in Figure 9.5, where the multidimensional schemata, OLAP queries, and semantic mappings at the OLAP level are translated to the relational model. As to multidimensional schemata, without loss of generality we assume that they are stored at the relational level as star schemata. As to queries, a classic logic-based syntax is adopted to express them at the relational level. As to mappings, their representation at the relational level uses a logical formalism typically adopted for schema mapping languages, i.e., *source-to-target tuple generating dependencies* (s-t tgd's) [11]. A query is then reformulated starting from its relational form on a star schema, using the mappings expressed as s-t tgd's.

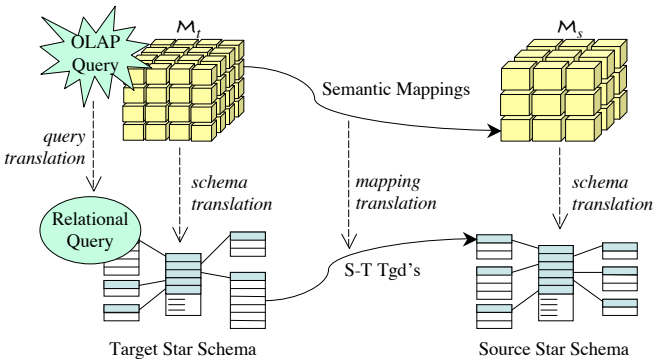

Fig. 9.5. A reformulation framework

A detailed explanation of the reformulation process can be found in [41]. Here we just provide an intuition based on a couple of examples. Note that, remarkably, all the translation steps outlined below are automated; the only manual phase in the whole process is the definition of the semantic mappings between each couple of neighboring peers.

Example 2. To start simple, consider the OLAP query q asking, at the London peer, for the total funding of projects about each subcategory of category 'Information Systems' on April 2011. Consistently with the framework sketched in Figure 9.5, reformulating this query onto the Rome peer requires:

1. Translating the multidimensional schemata at both London and Rome into star schemata, which produces the result shown in Figure 9.6.

@London
ProjectFT(coordinator,completed,ACMTopic,month,totalFunding,numberOfResUnits)
CoordinatorDT(coordinator,university,county,nation,position)
ACMTopicDT(ACMTopic,ACMSubcategory,ACMCategory)
MonthDT(month,year)

@Rome
FundingFT(researchUnit,project,month,amount)
ResearchUnitDT(researchUnit,university,city)
ProjectDT(project,subcategory,category,coordinatingUniv,city)

Fig. 9.6. Star schemata for the London and Rome peers

2. Translating q into a relational query on the London star schema:

$$q : \pi_{\mathsf{ACMSubcategory},\mathtt{SUM}(\mathsf{totalFunding})} \sigma_{(\mathsf{month}=\text{'April 2011'},\mathsf{ACMCategory}=\text{'Inf. Sys.'})} \chi_{London}$$

   where χ_{London} denotes the star join made over the London star schema.
3. Translating the mappings involved into s-t tgd's. For this query, the involved mappings are:

$$\mathsf{ACMCategory}\ \mathtt{equi\text{-}level}_f\ \mathsf{category}$$
$$\mathsf{ACMSubcategory}\ \mathtt{equi\text{-}level}_g\ \mathsf{subcategory}$$
$$\mathsf{month}\ \mathtt{equi\text{-}level}_h\ \mathsf{month}$$
$$\langle \mathsf{totalFunding}, \mathtt{SUM} \rangle\ \mathtt{same}_{expr}\ \mathsf{amount}$$

   where f and g are the identity function, h converts the Rome format for months ('04-11') into the London format ('April 2011'), and *expr* is $\mathsf{amount} * 0.872$.

Using the reformulation algorithm proposed in [41], q is then translated into the following query over the Rome schema:

$$q' : \pi_{\mathsf{subcategory},\mathtt{SUM}(\mathsf{amount}*0.872)} \sigma_{(h(\mathsf{month})=\text{'April 2011'},\mathsf{category}=\text{'Inf. Sys.'})} \chi_{Rome}$$

Remarkably, in this case reformulation is *compatible*, i.e., it fully preserves the semantics of q. When a compatible reformulation is used, the results returned by the source peer do *exactly* match with q so they can be seamlessly integrated with those returned by the target peer.

Example 3. Consider now the query asking, at the London peer, for the yearly funding of projects about topic 'Heterogeneous Databases':

$$q : \pi_{\mathsf{year},\mathtt{SUM}(\mathsf{totalFunding})}\sigma_{(\mathsf{ACMTopic}=\text{'Heterogeneous Databases'})}\chi\mathit{London}$$

In this case the mappings involved in reformulation are

$$\begin{aligned} \mathsf{ACMTopic}\ \mathtt{drill\text{-}down}_r\ \mathsf{subcategory} \\ \mathsf{year}\ \mathtt{roll\text{-}up}_s\ \mathsf{month} \\ \langle \mathsf{totalFunding}, \mathtt{SUM} \rangle\ \mathtt{same}_{expr}\ \mathsf{amount} \end{aligned}$$

where r is a function that associates each topic with its subcategory in the ACM classification, and s associates each month with its year. Query q is then translated into the following query over the Rome schema:

$$q' : \pi_{s(\mathsf{month}),\mathtt{SUM}(\mathsf{amount}*0.872)}\sigma_{(\mathsf{subcategory}=r(\text{'Heterogeneous Databases'}))}\chi\mathit{Rome}$$

Differently from Example 2, here reformulation is not compatible, so the results returned by the Rome peer match q with some approximation. In particular, since the topic detail is not present in the Rome schema, data are returned for the whole subcategory of 'Heterogeneous Databases' (i.e., 'DATABASE MANAGEMENT'). Although an integration with the London data is not possible in this case, users can still exploit the results coming from Rome, for example by comparing them with the London data at the finest common aggregation level (subcategory in this case).

See [41] for a discussion of the issues arising with compatible and non compatible reformulations.

9.6.3 Open Issues

A BIN is a complex system, whose construction and management requires sophisticated techniques, mostly devised to cope with the peculiarities of decision-making oriented information. In the following we outline the main issues to be solved to ensure that a BIN operates in a reliable, effective, and efficient way [42]:

- Answering queries in a BIN may be a very resource-consuming task both for the computational effort which is required to each queried peer and for the amount of exchanged messages. In order to avoid this, techniques for optimizing the reformulation process in the network must be adopted. In particular, *query routing strategies* should be used to prune redundant paths and forward queries to the most promising peers only, and distributed caching models should be together with online aggregation techniques to minimize query execution costs.

- A BIN must provide a unified, integrated vision of the heterogeneous information collected from the different peers to answer a user query. To this end, *object fusion* functionalities must be adopted to properly reconcile the multidimensional results returned; this task is made more complex by the fact that, due to heterogeneity of multidimensional schemata, the information returned may be not completely compliant with the original user query (e.g., it may have a different granularity).
- When a user performs a query, the other peers will often return results that are not exactly conformed to the schema of that query. For this reason, a BIN requires *smart user interfaces* capable of emphasizing the differences and relationships between the returned data, as well as techniques to rank the returned data depending on how compliant they are with the original local query.
- A BIN should include mechanisms for controlling *data provenance and quality* in order to provide users with information they can rely on. A mechanism for data lineage is also necessary to help users understand the semantics of the retrieved data and how these data have been transformed to handle heterogeneity.
- The nature of the exchanged information, as well as the presence of participants that belong to different organizations, require advanced approaches for *security*, ranging from proper access policies to data sharing policies that depend on the degree of trust between participants, as well as techniques for protecting against undesired information inference.

9.7 Conclusion

Extending the existing BI architectures and techniques to support collaborative scenarios is a challenging task that lays the foundations for BI 2.0. In this paper we have presented the benefits of collaborative BI and we have discussed the different approaches in the literature. Then we have outlined the BIN approach, that uses a P2P architecture to enable decentralized information sharing across a network of heterogeneous and autonomous peers.

As shown in Section 9.6.3, several interesting research directions can be explored to enrich and improve the BIN approach. Though most of them are strictly related to research issues already tackled for OLTP data in P2P systems, the solutions presented in that context do not adequately apply to the specific scenario of a BIN, because they do not effectively deal with the peculiarities of multidimensional data and OLAP query processing. For instance, the BIN mapping language could be effectively coupled with a language for expressing user preferences, aimed at better tailoring the reformulation process to the user's wishes. As a first step in this direction, the specification of mappings can be enriched with a similarity score to express the semantic correlation of the source and target sets of concepts. Similarity may depend on the differences in the peers' vocabularies as well as on different perspectives of data representation (e.g., different granularities), and should be influenced

by the absence or presence of transcoding functions that make source and target values compatible. Such a score, representative of the semantic strength of a mapping, can be profitably employed in query reformulation where, in presence of alternative mappings for a given set of concepts onto a source peer, it can be used to identify the mapping that best approximates this set of concepts, so as to translate the query as accurately as possible. Then, the similarity scores of the mappings involved in the reformulation process can be combined to determine how compliant the results obtained from a source peer are, overall, with respect to the original query [3]. This compliance score can be profitably used by users to rank (and, possibly, filter) the results obtained from different source peers.

References

1. Lachlan, J.: Top 13 business intelligence trends for (2011), `http://www.japan.yellowfin.bi`
2. Hoang, T.A.D., Nguyen, T.B.: State of the art and emerging rule-driven perspectives towards service-based business process interoperability. In: Proc. Int. Conf. on Comp, and Comm. Tech., Danang City, Vietnam, pp. 1–4 (2009)
3. Banek, M., Vrdoljak, B., Tjoa, A.M., Skocir, Z.: Automated integration of heterogeneous data warehouse schemas. IJDWM 4(4), 1–21 (2008)
4. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: Semantics and query answering. In: Proc. ICDT, pp. 207–224 (2003)
5. Lenzerini, M.: Data integration: A theoretical perspective. In: Proc. PODS, pp. 233–246 (2002)
6. Halevy, A.Y.: Technical perspective - schema mappings: rules for mixing data. Commun. ACM 53(1) (2010)
7. Halevy, A.Y., Ives, Z.G., Madhavan, J., Mork, P., Suciu, D., Tatarinov, I.: The Piazza peer data management system. IEEE TKDE 16(7), 787–798 (2004)
8. Fuxman, A., Kolaitis, P.G., Miller, R.J., Tan, W.C.: Peer data exchange. In: Proc. PODS, pp. 160–171 (2005)
9. Tatarinov, I., Halevy, A.Y.: Efficient query reformulation in peer-data management systems. In: Proc. SIGMOD Conf., Paris, France, pp. 539–550 (2004)
10. Halevy, A.: Answering queries using views: A survey. VLDB Journal 10(4), 270–294 (2001)
11. ten Cate, B., Kolaitis, P.G.: Structural characterizations of schema-mapping languages. Commun. ACM 53(1), 101–110 (2010)
12. Cudré-Mauroux, P., Aberer, K., Feher, A.: Probabilistic message passing in peer data management systems. In: Proc. ICDE, Atlanta, USA, p. 41 (2006)
13. Madhavan, J., Bernstein, P.A., Doan, A., Halevy, A.Y.: Corpus-based schema matching. In: Proc. ICDE, Tokyo, Japan, pp. 57–68 (2005)
14. Mecca, G., Papotti, P., Raunich, S.: Core schema mappings. In: Proc. SIGMOD, pp. 655–668 (2009)
15. Fagin, R., Kolaitis, P.G., Popa, L.: Data exchange: getting to the core. ACM Trans. Database Syst. 30(1), 174–210 (2005)
16. Chang, K.C., Garcia-Molina, H.: Mind your vocabulary: Query mapping across heterogeneous information sources. In: Proc. SIGMOD, pp. 335–346 (1999)

17. Papakonstantinou, Y., Abiteboul, S., Garcia-Molina, H.: Object fusion in mediator systems. In: Proc. VLDB, Bombay, India, pp. 413–424 (1996)
18. Torlone, R.: Two approaches to the integration of heterogeneous data warehouses. Distributed and Parallel Databases 23(1), 69–97 (2008)
19. Jiang, H., Gao, D., Li, W.S.: Exploiting correlation and parallelism of materialized-view recommendation for distributed data warehouses. In: Proc. ICDE, Istanbul, Turkey, pp. 276–285 (2007)
20. Berger, S., Schrefl, M.: From federated databases to a federated data warehouse system. In: Proc. HICSS, Waikoloa, Big Island of Hawaii, p. 394 (2008)
21. Jindal, R., Acharya, A.: Federated data warehouse architecture (2004), http://www.wipro.com/
22. Albrecht, J., Lehner, W.: On-line analytical processing in distributed data warehouses. In: Proc. IDEAS, pp. 78–85 (1998)
23. Akinde, M.O., Böhlen, M.H., Johnson, T., Lakshmanan, L.V.S., Srivastava, D.: Efficient OLAP query processing in distributed data warehouses. Inf. Syst. 28(1-2), 111–135 (2003)
24. Banek, M., Tjoa, A.M., Stolba, N.: Integrating Different Grain Levels in a Medical Data Warehouse Federation. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2006. LNCS, vol. 4081, pp. 185–194. Springer, Heidelberg (2006)
25. Schneider, M.: Integrated vision of federated data warehouses. In: Proc. DISWEB, Luxemburg (2006)
26. Berger, S., Schrefl, M.: Analysing Multi-Dimensional Data across Autonomous Data Warehouses. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2006. LNCS, vol. 4081, pp. 120–133. Springer, Heidelberg (2006)
27. Mangisengi, O., Huber, J., Hawel, C., Eßmayr, W.: A Framework for Supporting Interoperability of Data Warehouse Islands using XML. In: Kambayashi, Y., Winiwarter, W., Arikawa, M. (eds.) DaWaK 2001. LNCS, vol. 2114, pp. 328–338. Springer, Heidelberg (2001)
28. Tseng, F.S.C., Chen, C.W.: Integrating heterogeneous data warehouses using XML technologies. J. Information Science 31(3), 209–229 (2005)
29. Bruckner, R.M., Ling, T.W., Mangisengi, O., Tjoa, A.M.: A framework for a multidimensional OLAP model using topic maps. In: Proc. WISE (2), pp. 109–118 (2001)
30. Zhou, S., Zhou, A., Tao, X., Hu, Y.: Hierarchically distributed data warehouse. In: Proc. HPC, Washington, DC, pp. 848–853 (2000)
31. Abiteboul, S.: Managing an XML Warehouse in a P2P Context. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681, pp. 4–13. Springer, Heidelberg (2003)
32. Abiteboul, S., Manolescu, I., Preda, N.: Constructing and querying peer-to-peer warehouses of XML resources. In: Proc. SWDB, Toronto, Canada, pp. 219–225 (2004)
33. Bonifati, A., Chang, E.Q., Ho, T., Lakshmanan, L.V.S., Pottinger, R., Chung, Y.: Schema mapping and query translation in heterogeneous P2P XML databases. VLDB J 19(2), 231–256 (2010)
34. Espil, M.M., Vaisman, A.A.: Aggregate queries in peer-to-peer OLAP. In: DOLAP, Washington, DC, USA, pp. 102–111 (2004)
35. Vaisman, A., Espil, M.M., Paradela, M.: P2P OLAP: Data model, implementation and case study. Information Systems 34(2), 231–257 (2009)

36. Kehlenbeck, M., Breitner, M.H.: Ontology-Based Exchange and Immediate Application of Business Calculation Definitions for Online Analytical Processing. In: Pedersen, T.B., Mohania, M.K., Tjoa, A.M. (eds.) DaWaK 2009. LNCS, vol. 5691, pp. 298–311. Springer, Heidelberg (2009)
37. Kalnis, P., Ng, W.S., Ooi, B.C., Papadias, D., Tan, K.L.: An adaptive peer-to-peer network for distributed caching of OLAP results. In: Proc. SIGMOD, Madison, Wisconsin, pp. 25–36 (2002)
38. Dubois, D., Prade, H.: On the use of aggregation operations in information fusion processes. Fuzzy Sets and Systems 142(1) (2004)
39. Golfarelli, M., Mandreoli, F., Penzo, W., Rizzi, S., Turricchia, E.: Towards OLAP query reformulation in peer-to-peer data warehousing. In: Proc. DOLAP, pp. 37–44 (2010)
40. Golfarelli, M., Rizzi, S.: Data Warehouse design: Modern principles and methodologies. McGraw-Hill (2009)
41. Golfarelli, M., Mandreoli, F., Penzo, W., Rizzi, S., Turricchia, E.: OLAP query reformulation in peer-to-peer data warehousing. Information Systems (to appear, 2011)
42. Golfarelli, M., Mandreoli, F., Penzo, W., Rizzi, S., Turricchia, E.: BIN: Business intelligence networks. In: Business Intelligence Applications and the Web: Models, Systems and Technologies. IGI Global (2011)