A Query Reformulation Framework for P2P OLAP*

Matteo Golfarelli[1], Federica Mandreoli[2],
Wilma Penzo[1], Stefano Rizzi[1], and Elisa Turricchia[1]

[1] DEIS - Univ. of Bologna, V.le Risorgimento 2, Bologna, Italy
[2] DII - Univ. of Modena and Reggio Emilia, Via Vignolese 905/b, Modena, Italy

Abstract. The idea of collaborative business intelligence is to extend the decision-making process beyond the company boundaries thanks to cooperation and data sharing with other companies and organizations. In this direction, we propose a query reformulation framework based on a P2P network of heterogeneous peers, each exposing OLAP query answering functionalities aimed at sharing business information. In our framework, an OLAP query expressed on a peer is reformulated on other peers by relying on a set of mappings between the multidimensional schemata of peers. In this extended abstract we sketch the user interaction scenario we envision and briefly discuss each phase of the reformulation process.

Keywords: business intelligence, OLAP, P2P architectures, query reformulation

1 Introduction

In the current changeable and unpredictable market scenarios, the needs of decision makers are rapidly evolving as well. This gave rise to a new generation of business intelligence (BI) systems, often labeled as *BI 2.0*. One of the key features of BI 2.0 is the ability to become collaborative and extend the decision-making process beyond the boundaries of a single company [12]. Users need to transparently 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. For instance, this is the case of companies in a supply chain, or local health-care departments that collaborate to enable effective analysis of epidemics and health-care costs [6]. In such a complex and distributed business scenario, traditional BI systems —that were born to support stand-alone decision-making— are no longer sufficient to maximize the effectiveness of monitoring and decision making processes.

To fill this gap, we envision a peer-to-peer data warehousing architecture called *Business Intelligence Network* (BIN). A BIN is an architecture for sharing

* An extended version of this work is published in [6].

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

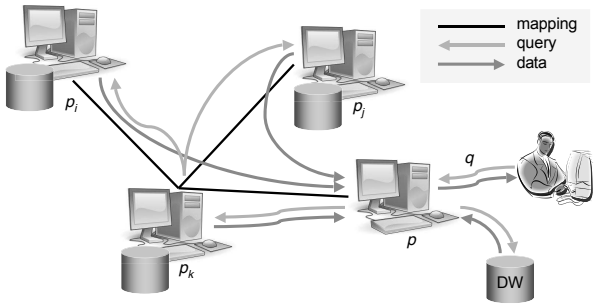

Fig. 1. Interaction scenario for a BIN

BI functionalities across a dynamic and collaborative network of heterogeneous and autonomous peers. Each peer is equipped with an independent data warehouse system, that relies on a local multidimensional schema to represent the peer's view of the business and exposes OLAP query answering functionalities aimed at sharing business information, in order to enhance the decision making process and create new knowledge. The main benefit of the BIN approach stands in the ability to efficiently manage inter-company processes and safely sharing management information besides operational information [5].

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 (Figure 1):

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 (*target*) peer.

In line with the approach adopted in *Peer Data Management Systems* (PDMSs) [8], 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 source peer map onto those of the target peer. Peers establish semantic mappings by exchanging their local schemata and applying a schema-matching algorithm [11]. 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 the neighborhood of p; then, each peer in

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

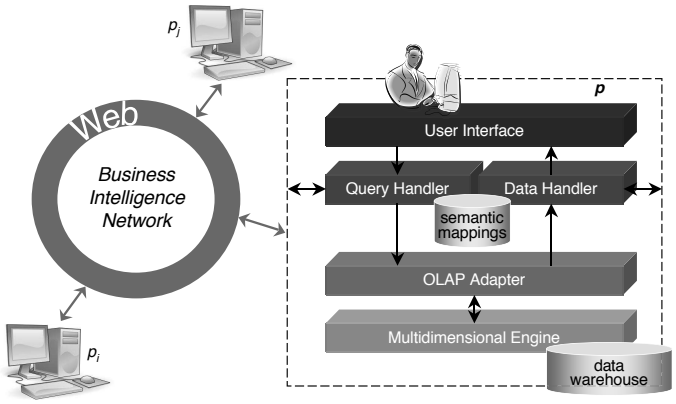

Fig. 2. Envisioned architecture for a BIN

this neighborhood in turn sends q to its neighborhood, and so on.[3] 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.

The approach outlined above is reflected by the internal architecture of each peer, sketched in Figure 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 neighborhood (using the available semantic mappings), and transmits it to the peers in that neighborhood.
3. *Data Handler*. When processing a locally-formulated query, the data handler collects query results from the OLAP adapter and from the other peers, integrates them, and returns them to the user interface. When processing a query formulated on some other peer p, the data handler just collects local query results from the OLAP adapter and returns them to 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.

Query answering in a BIN architecture poses several research challenges, ranging from languages and models for semantic mediation to query reformulation issues and proper techniques and data structures for the query processing

[3] To improve query processing efficiency, query routing strategies to select a subset of neighboring peers for query reformulation could be employed [11].

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

phase. In particular, much work on query reformulation has been done in the context of PDMSs [8] and relational databases [4], however those results are not directly applicable in the OLAP scenario presented by the BIN, that poses additional challenges due to the presence of aggregation and to the possibility of having information represented at different granularities in each peer. The framework for query reformulation in a BIN we outline in this extended abstract relies on the translation of semantic mappings and queries towards the underlying relational schemata. Mappings between the schemata of peers are expressed using predicates that are specifically tailored for the multidimensional model; to overcome possible differences in data formats and information granularities, mappings can be associated with transcoding functions. The query reformulation algorithm is correct, with polynomial complexity [6], and can be safely used to implement chains of reformulations as required in the BIN setting.

In the following sections the main aspects of the query reformulation process will be intuitively discussed based on a working example.

2 Mapping Language

Reformulation of OLAP queries first of all requires a language for properly expressing the semantic mappings between each couple of neighboring peers. The language used in a BIN accommodates the peculiar characteristics of the multidimensional model, on which the representation of business information at each peer is founded. It expresses 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 using the *mapping predicates* 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. A transcoding function can be a standard database function (e.g., `substring`) shared by all peers, as well as a function owned by a peer and made available to its neighbors by attaching it to query messages. If this function is available, it is used to increase the reformulation effectiveness, e.g., by enabling data returned by the source and target peers to be integrated.

- The `same` predicate states that whenever a given measure is asked in a query on $\mathcal{M}_t$ using a given aggregation operator, it can be rewritten as a given expression involving the measures in $\mathcal{M}_s$.
- The `equi-level` predicate states that two sets of attributes of $\mathcal{M}_t$ and $\mathcal{M}_s$, respectively, have the same semantics and granularity. Optionally, it can be annotated with a transcoding function that establishes a one-to-one relation between tuples of values of the two sets of attributes.
- The `roll-up` predicate states that a set of attributes of $\mathcal{M}_t$ aggregates a set of attributes of $\mathcal{M}_s$. Optionally, it can be annotated with a transcoding

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

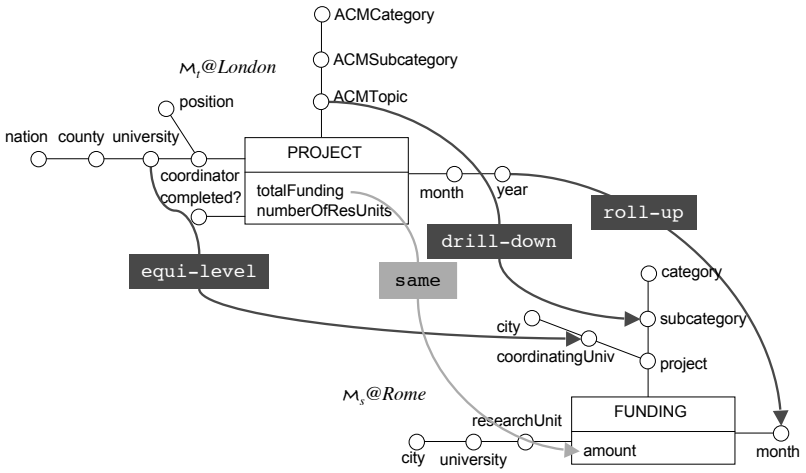

Fig. 3. Multidimensional schemata of related facts at two peers

function that establishes a many-to-one relation between tuples of values of the two sets of attributes.

– The `drill-down` predicate states that a set of attributes of $\mathcal{M}_t$ disaggregates a set of attributes of $\mathcal{M}_s$. Optionally, it can be annotated with a transcoding function that establishes a one-to-many relation between tuples of values of the two sets of attributes.

Example 1. Consider a BIN for sharing information about funded research projects among European nations. Figure 3 shows the multidimensional schemata of related facts at the peers in London and Rome, using the Dimensional Fact Model notation [7]; small circles represent attributes, while measures are listed inside the fact boxes. Figure 3 also shows some of the mappings that can be 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 using the exchange rate 0.872. □

3 Inter-Peer 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 adopt is based on a relational setting where the multidimensional schemata, OLAP queries, and semantic mappings at the OLAP level are translated to the relational model. As to multidimensional schemata,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

@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. 4. Star schemata for the London and Rome peers

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) [3]. A query is then reformulated starting from its relational form on a star schema, using the mappings expressed as s-t tgd's. A detailed explanation of the reformulation process can be found in [6], together with the reformulation algorithm and its proof of correctness.

Example 2. Consider the OLAP query q asking, at the London peer, for the total funding of projects about each subcategory of category 'Information Systems' in 2011. 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 4.
2. Translating q into a relational query on the London star schema:

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

   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 (each annotated with its transcoding or expression) are:

$$\begin{aligned} \mathsf{ACMCategory}\ \mathtt{equi\text{-}level}_{id}\ \mathsf{category} \\ \mathsf{ACMSubcategory}\ \mathtt{equi\text{-}level}_{id}\ \mathsf{subcategory} \\ \mathsf{year}\ \mathtt{roll\text{-}up}_{\mathtt{YearOf}}\ \mathsf{month} \\ \langle \mathsf{totalFunding}, \mathtt{SUM} \rangle\ \mathtt{same}_{\mathsf{amount}*0.872}\ \mathsf{amount} \end{aligned}$$

   where id is the identity function and the `YearOf` transcoding associates each month in the Rome format to its year in the London format.

Using the reformulation algorithm proposed in [6], q is then translated into the following query over the Rome schema:

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

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. □

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

4 Intra-Peer Reformulation

In general, a BIN query (either directly formulated by a user or reformulated across the network) cannot be directly executed on the peer local multidimensional engine, because of the language and expressiveness gap between the query handler and the local multidimensional engine. The OLAP adapter is in charge of bridging this gap by supporting intra-peer reformulation of BIN queries, so as to complete the reformulation process. Assuming that the de-facto standard MDX is the querying language of the local multidimensional engine, intra-peer reformulation must deal with the presence of transcodings in the query group-by set, and must properly manage non-distributive aggregation operators. From the reformulation point of view, this amounts to solving a problem of *query rewriting using views* [9], where the set of views is made of all the possible queries that the engine supports. In particular, given the relational translation q of a BIN query, we have to find a local query q^{loc} that refers to one MDX result set and is equivalent to q.

Example 3. The query q' shown in Example 2 cannot be directly formulated in MDX because it involves the `YearOf` transcoding. The SQL for the corresponding local query is

```
SELECT      RS.subcategory, SUM(RS.amount*0.872)
FROM        ResultSet RS, YearOf YO
WHERE       RS.month = YO.month AND YO.year = '2011'
GROUP BY    RS.subcategory;
```

where YearOf is a lookup table for the `YearOf` transcoding and ResultSet stores the result of the following MDX query:

```
SELECT {[Measures].[amount]} ON COLUMN,
       {NonEmptyCrossJoin([Month].[month].Members,
                          [Project].[Inf.Sys.].Children)} ON ROWS
FROM [Funding]
```

□

5 Conclusions and Related Works

Supporting the sharing of information for decision-making processes is a challenging task that lays the foundations for BI 2.0. The BIN architecture and the query reformulation framework we proposed is a first, significant step in this direction. Noticeably, despite the relevance of the problem, only a few works in the literature are specifically focused on strategies for data warehouse integration and federation. Indeed, in this context, problems related to data heterogeneity are usually solved by ETL processes that read data from several data sources and load them in a single repository. While this centralized architecture may fit the needs of stand-alone companies, it is hardly feasible in the context of a BIN, where the dynamic nature of the business network, together with the independence and autonomy of peers, call for more sophisticated solutions. See [1] for

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

a discussion of the benefits of a peer-to-peer architecture for data warehousing and [13] for a description of the issues arising in collaborative BI systems.

In the context of a federated data warehouse architecture, [14] describes two methods for integrating dimensions belonging to different data marts, but the problem of how to define mappings between concepts is not considered. The work proposed in [2] presents a complete algorithm for matching multidimensional structures, but the data-related aspects are not considered, and no model is provided to formalize the mapping predicates. Another work centered on interoperability issues is [10]; since it proposes specific techniques to deal with measures only, it cannot be used to completely solve a typical aggregate query. The work that is most closely related to ours is [15]; though some goals are shared with our approach, there are important differences: peers' autonomy is not preserved and the problem of chains of reformulation is not faced.

References

1. Abiteboul, S.: Managing an XML warehouse in a P2P context. In: Proc. CAiSE. pp. 4–13. Klagenfurt, Austria (2003)
2. Banek, M., Vrdoljak, B., Tjoa, A.M., Skocir, Z.: Automated integration of heterogeneous data warehouse schemas. IJDWM 4(4), 1–21 (2008)
3. ten Cate, B., Kolaitis, P.G.: Structural characterizations of schema-mapping languages. Commun. ACM 53(1), 101–110 (2010)
4. Cohen, S., Nutt, W., Sagiv, Y.: Rewriting queries with arbitrary aggregation functions using views. TODS 31(2), 672–715 (2006)
5. 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, pp. 244–265. IGI Global (2011)
6. Golfarelli, M., Mandreoli, F., Penzo, W., Rizzi, S., Turricchia, E.: OLAP query reformulation in peer-to-peer data warehousing. Inf. Syst. 37(5), 393–411 (2012)
7. Golfarelli, M., Rizzi, S.: Data Warehouse design: Modern principles and methodologies. McGraw-Hill (2009)
8. 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)
9. Halevy, A.: Answering queries using views: A survey. VLDBJ 10(4), 270–294 (2001)
10. Kehlenbeck, M., Breitner, M.H.: Ontology-based exchange and immediate application of business calculation definitions for online analytical processing. In: Proc. DaWaK. pp. 298–311. Linz, Austria (2009)
11. Mandreoli, F., Martoglia, R., Penzo, W., Sassatelli, S.: Data-sharing P2P networks with semantic approximation capabilities. IEEE Internet Computing 13(5), 60–70 (2009)
12. Raden, N.: Business intelligence 2.0: Simpler, more accessible, inevitable. http://intelligent-enterprise. informationweek.com (2007)
13. Rizzi, S.: Collaborative business intelligence. In: Business Intelligence - First European Summer School, pp. 186–205. Springer (2012)
14. Torlone, R.: Two approaches to the integration of heterogeneous data warehouses. Distributed and Parallel Databases 23(1), 69–97 (2008)
15. Vaisman, A., Espil, M.M., Paradela, M.: P2P OLAP: Data model, implementation and case study. Inf. Syst. 34(2), 231–257 (2009)

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,