# Grid Data Integration based on Schema-mapping

Carmela Comito and Domenico Talia

DEIS, University of Calabria,
Via P. Bucci 41 c,
87036 Rende, Italy
{ccomito, talia}@deis.unical.it
http://www.deis.unical.it/

**Abstract.** Data integration is the flexible and managed federation, analysis, and processing of data from different distributed sources. Data integration is a key issue for exploiting the availability of large, heterogeneous, distributed and highly dynamic data volumes on Grids. This paper presents a framework for integrating heterogeneous XML data sources distributed among the nodes of a Grid. We present a query reformulation algorithm to combine and query XML documents through a decentralized point-to-point mediation process among the different data sources based on schema mappings. The above cited XML integration formalism is exposed as a Grid Service within an OGSA-based Grid architecture.

## 1 Introduction

The huge amount of data today available in a digital format requires solutions for their integrated access and analysis. The goal of data integration systems is to combine heterogeneous data residing at different sites by providing a unified view of this data.

Data integration on Grids has to deal with unpredictable, highly dynamic data volumes. So, traditional approaches to data integration, such as FDBMS [1] and the use of mediator/wrapper middleware [2], are not suitable in Grid settings. The federation approach is a rather rigid configuration where resources allocation is static and optimization cannot take advantage of evolving circumstances in the execution environment. The design of mediator/wrapper integration systems must be done globally and the coordination of mediators has to be done centrally, which is an obstacle to the exploitation of evolving characteristics of dynamic environments. As a consequence, data sources cannot change often and significantly, otherwise they may violate the mappings to the mediated schema.

Recently, several works on data management in peer-to-peer (P2P) systems are moving along this direction [3, 4]. All these systems focus on an integration approach not based on a global schema: each peer represents an autonomous information system, and data integration is achieved by establishing mappings among the various peers.

The Grid community is devoting great attention toward the management of structured and semi-structured data such as databases and XML data. The most significant examples of such efforts are the *OGSA Data Access and Integration*

(OGSA-DAI) [5] and the *OGSA Distributed Query Processor* (OGSA-DQP) [6] projects. However, till today only few of those projects [7,8] actually meet schema-integration issues necessary for establishing semantic connections among heterogeneous data sources. For these reasons, we designed the XMAP framework for integrating heterogeneous XML data sources distributed over a Grid. By designing this framework, we aim at developing a decentralized network of semantically related schemas that enables the formulation of distributed queries over heterogeneous data sources. We designed a method to combine and query XML documents through a decentralized point-to-point mediation process among the different data sources based on schema mappings. We offer a decentralized service-based architecture that exposes this XML integration formalism as a Grid Service [9]. We refer to this architecture as the *Grid Data Integration System* (GDIS). The GDIS infrastructure exploits the middleware provided by OGSA-DQP, OGSA-DAI, and Globus Toolkit 3 [10], building on top of them schema-integration services.

## 2 XMAP: A Decentralized XML Data Integration Framework

In this section, we briefly describe the XMAP framework [11], a decentralized network of semantically related XML schemas that enables the formulation of queries over heterogeneous, distributed data sources. The environment is modeled as a system composed of a number of Grid nodes, where each node can hold one or more XML databases. These nodes are connected to each other through declarative mappings rules.

### 2.1 Integration Model

Our integration model is based on schema mappings to translate queries between different schemas. The goal of a schema mapping is to capture structural as well as terminological correspondences between schemas.

As mentioned before, traditional centralized architecture of data integration systems is not suitable for highly dynamic and distributed environments such as the Grid. Thus, we propose an approach inspired from [4] where the mapping rules are established directly among source schemas without relying on a central mediator or a hierarchy of mediators. In consequence, in our integration model, there is no global schema representing all data sources in a unique data model but a collection of local schemas (the native schema of each data source).

The specification of mappings is thus flexible and scalable. Regardless of the total number of nodes composing the system, each source schema is directly connected to only a small number of other schemas. However, it remains reachable from all other schemas that belong to its “transitive closure”. For any mapping M, its closure is defined as the set of rules that can be derived from M by repeated composition of schema paths. In other words, the system supports two different kinds of mapping to connect schemas semantically: *point-to-point* mappings and *transitive* mappings. In transitive mappings, data sources are related through one or more “mediator schemas”. For example, if we have a source A directly connected to a source B and B connected to C, A is connected to *both* B and C. Establishing the mappings this way creates a graph of semantically related sources where each of the sources knows its direct semantic neighbors (point-to-point mapping) and can learn about the mappings of its neighbors (transitive mapping). Therefore, in our integration model all nodes are equal: there is no distinction between data sources and mediators. Each node acts both as a data source contributing data and as a local mediator providing an uniform view over the data provided by other nodes.

We address structural heterogeneity among XML data sources by associating paths in different schemas. Mappings are specified as path expressions that relate a specific element or attribute (together with its path) in the source schema to related elements or attributes in the destination schema. The data integration model we propose is indeed based on path-to-path mappings expressed in the XPath [12] query language, assuming XML Schema as the data model for XML sources. Specifically, this means that a path in a source is described in terms of XPath expressions.

The mapping rules are specified in XML documents called XMAP documents. Each source schema in the framework is associated to an XMAP document containing all the mapping rules related to it.

### 2.2 The XMAP Reformulation Algorithm

In this section we present an algorithm to reformulate an XPath query on the basis of the mapping rules established for the schema over which the query is formulated. In the following, we suppose that we have a set of XML data sources, that each data source is compliant to an XML Schema and that, for each schema, an XMAP document containing the mappings related to this schema is provided.

Our query processing approach exploits the semantic connections established in the system by performing the *query reformulation algorithm* before executing the query, in order to gain further knowledge. This way, when a query is posed over the schema of a source, the system will be able to use data from any source that is transitively connected by semantic mappings. Indeed, it will reformulate the given query expanding and translating it into appropriate queries for each semantically related source. Thus, the user can retrieve data from all the related sources in the system by simply submitting a single XPath query.

The query reformulation algorithm uses as input an XPath query and the mappings, and it produces as output zero, one or more reformulated queries. We describe now in details the logic of the algorithm. In this discussion, we use $Q$ to denote the input XPath query, $S$ the source schema over which $Q$ is formulated, $M$ the mappings in the system and $Q_{R_i}$ the reformulated queries produced by the algorithm.

Figure 1 describes an example of use of the XMAP algorithm. Here a query $Q$ is formulated over the schema $S_1$. In the first step the algorithm identifies the paths $P_1$ and $P_2$ in $Q$ and produces as output the set $\mathcal{P}$. Next, exploiting the XMAP document associated to the schema $S_1$, the algorithm finds two mapping rules connecting $S_1$ to $S_2$ trough the paths $P_1$ and $P_2$. More precisely, one of these

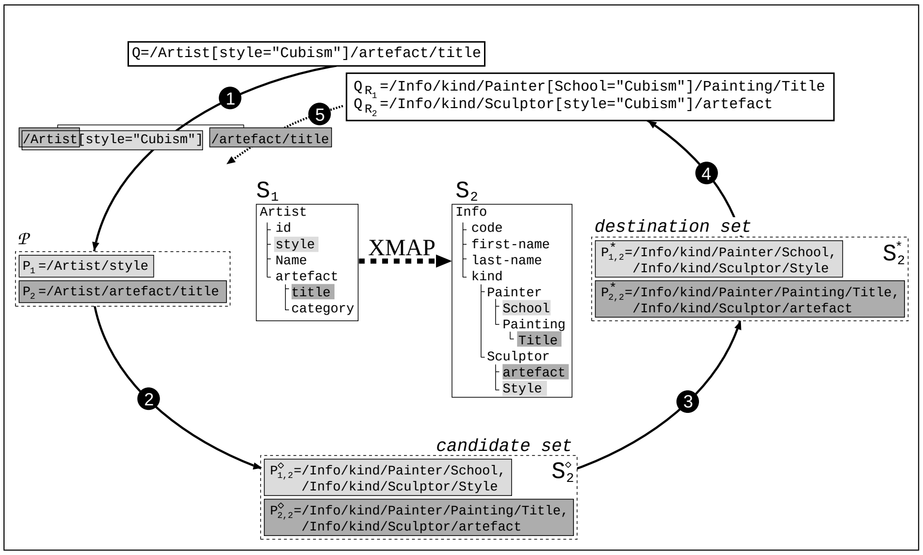
Q=/Artist[style="Cubism"]/artefact/title
1
Q$_{R_1}$ =/Info/kind/Painter[School="Cubism"]/Painting/Title
Q$_{R_2}$ =/Info/kind/Sculptor[style="Cubism"]/artefact
5
/Artist[style="Cubism"]
/artefact/title
4
S$_1$
Artist
id
style
Name
artefact
title
category
XMAP
S$_2$
Info
code
first-name
last-name
kind
Painter
School
Painting
Title
Sculptor
artefact
Style
$\mathcal{P}$
P$_1$ =/Artist/style
P$_2$ =/Artist/artefact/title
destination set
P$^*_{1,2}$=/Info/kind/Painter/School, /Info/kind/Sculptor/Style
P$^*_{2,2}$=/Info/kind/Painter/Painting/Title, /Info/kind/Sculptor/artefact
S$^*_2$
2
3
candidate set
P$^\diamond_{1,2}$=/Info/kind/Painter/School, /Info/kind/Sculptor/Style
P$^\diamond_{2,2}$=/Info/kind/Painter/Painting/Title, /Info/kind/Sculptor/artefact
S$^\diamond_2$

**Fig. 1.** Example of use of the XMAP algorithm.

rules relates P1 to two paths in $S_2$, respectively `/Info/kind/Painter/School` and `/Info/Kind/Sculptor/Artefact`. Similarly, the other mapping rules relates $P_2$ to `/Info/Kind/Painter/Painting/Title` and `/Info/Kind/Sculptor/artefact`. So, the second step of the algorithm produces as output a candidate set composed of the elements $P^\diamond_{i,j}$ and the (candidate) schema $S^\diamond_2$. In the considered example as the schema $S^\diamond_2$ has correspondences for both paths $P_1$ and $P_2$, it is identified as a destination schema, so it can be used to reformulate the query $Q$. In particular, the algorithm produces two reformulations of the query $Q$ over the schema $S_2$, respectively $Q_{R_1}$ and $Q_{R_2}$.

## 3 The Grid Data Integration System (GDIS)

The XMAP reformulation algorithm has been deployed and used in the *Grid Data Integration System* (GDIS). GDIS is a decentralized service-based data integration architecture for Grid databases; it has been extensively described in [13].

The GDIS system offers a wrapper/mediator-based approach to integrate data sources: it adopts the XMAP decentralized mediator approach to handle semantic heterogeneity over data sources, whereas syntactic heterogeneity is hidden behind ad-hoc wrappers. In the GDIS architecture (see [13]), the query reformulation engine is run by the *data integration nodes*. Specifically, these nodes offer: (i) a set of data integration utilities allowing to establish mappings, and (ii) the query reformulation algorithm introduced by the XMAP integration formalism. These utilities are exposed through the portTypes of the proposed OGSA-GDI data integration service.

GDIS is designed as a service-based distributed architecture where each node exposes data resources and data integration facility as Grid Data Services (GDSs) [14]. In so doing, the GDIS system introduces the OGSA-based *Grid Data Integration* (GDI) service that extends OGSA-DAI and OGSA-DQP portTypes with additional functionality both to enable users to specify semantic mappings (in the form of XMAP documents) among a set of data sources, and to execute the XMAP query rewriting algorithm. Among the portTypes introduced by GDI the most important is the *Query Reformulation Algorithm* (QRA) portType, that performs the XMAP query reformulation algorithm and receives as input a query and the schema mappings and produces as output a set of reformulated queries. The reformulated queries will be forwarded to the *Grid Data Service* portType offered by the OGSA-DQP distributed query service (GDQS).

## References

1. Sheth, A.P., Larson, J.A.: Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys **22** (1990) 183–236
2. Levy, A.Y., Rajaraman, A., Ordille, J.J.: Querying heterogeneous information sources using source descriptions. In: VLDB. (1996) 251–262
3. Calvanese, D., Damaggio, E., Giacomo, G.D., Lenzerini, M., Rosati, R.: Semantic data integration in P2P systems. In: DBISP2P. (2003) 77–90
4. Halevy, A.Y., Suciu, D., Tatarinov, I., Ives, Z.G.: Schema mediation in peer data management systems. In: ICDE. (2003) 505–516
5. Antonioletti, M., et al.: OGSA-DAI: Two years on. In: Global Grid Forum 10 — Data Area Workshop. (2004) `http://www.ogsadai.org.uk/`.
6. Alpdemir, M.N., Mukherjee, A., Gounaris, A., Paton, N.W., Watson, P., Fernandes, A.A.A., Fitzgerald, D.J.: OGSA-DQP: A service for distributed querying on the grid. In: EDBT. (2004) 858–861
7. Calvanese, D., Giacomo, G.D., Lenzerini, M., Rosati, R., Vetere, G.: Hyper: A framework for peer-to-peer data integration on grids. In: ICSNW. (2004) 144–157
8. Brezany, P., Woehrer, A., Tjoa, A.M.: Novel mediator architectures for grid information systems. FGCS - Grid Computing: Theory, Methods and Applications. **21** (2005) 107–114
9. Foster, I., Kesselman, C., Nick, J.M., Tuecke, S.: The physiology of the grid: An open grid services architecture for distributed systems integration. Open Grid Service Infrastructure WG, Global Grid Forum (2002) `http://www.globus.org/research/papers/ogsa.pdf`.
10. Sandholm, T., Gawor, J.: Globus toolkit 3 core — A grid service container framework. Globus Toolkit Core White Paper (2003) `http://www-unix.globus.org/toolkit/3.0/ogsa/docs/gt3_core.pdf`.
11. Comito, C., Talia, D.: XML data integration in OGSA grids. In: Proceedings of the First VLDB Workshop on Data Management in Grids, LNCS 3836. (2005)
12. Clark, J., DeRose, S.: XML path language (XPath) version 1.0. W3C Recommendation (1999) `http://www.w3.org/TR/xpath`.
13. Comito, C., Talia, D.: GDIS: A service-based architecture for data integration on grids. In: GADA. (2004) 88–98
14. Foster, I., Tuecke, S., Unger, J.: OGSA data services. DAIS-WG Informational Draft, 9th Global Grid Forum (2003)