Inter Model Data Integration in a P2P Environment

Duc Minh Le, Andrew Smith and Peter McBrien

Department of Computing, Imperial College London
180 Queens Gate, London SW7 2AZ
{dmle,acs203,pjm}@doc.ic.ac.uk

Abstract. The wide range of data sources available today means that the integration of heterogeneous data sources is now a common and important problem. It is even more challenging in a P2P environment where peers often do not know in advance which schemas of other peers will suit their information needs and there is potentially a greater diversity of data modelling languages in use. In this paper, we propose a new approach to P2P inter model data integration which supports multiple data models whilst allowing peers the flexibility of choosing how to integrate their schemas.

1 Introduction

P2P inter model data integration is the process whereby data stored in autonomous heterogeneous data sources under different data models is made accessible to other peers on a P2P network. There are three aspects to this. The first is how to represent the heterogeneous data sources in a **common data model (CDM)** [1] that can accurately represent the constructs of the various data models involved. The second is how to integrate these schemas to allow easy access to the relevant data sources. Finally we need a way of enabling peers to find the schemas created in the previous step. The difficulty is in handling increasing numbers of partially or fully integrated schemas on the network as schemas may be integrated by peers in an add-hoc fashion.

In this paper we propose a method of performing inter model data integration in a P2P environment by using the **Hypergraph Data Model(HDM)** [2] as a CDM and the **Both-as-View(BAV)** data integration method. We define a general framework, independent of the data models of the peer data sources, for representing schema metadata and for managing these metadata on the P2P network. We also formally define schema search and propose a distributed algorithm for searching for the relevant public schemas.

2 Representing and Transforming Schemas

In a P2P environment, peer data sources are often highly heterogeneous not only at the data but also at the metadata level. To combine these heterogeneous data sources a generic data model that is capable of expressing all the constructs of the different schemas is required. In this paper we use the HDM which has been used to represent a wide variety of data models [3] making it particularly suited as a CDM in inter model data integration. Figures 2 and 4 show a SQL database and an XML document represented in the HDM.

The HDM is a graph based model that makes use of 3 simple constructs: **nodes**, **edges** and **constraints**. The nodes and edges of an HDM schema, shown as circles and lines in the figures, represent the structure of the data source. We refer to nodes and edges collectively as **schema objects**. Any constraints on the data are represented using the HDM's generic constraint operators [2], shown in the figures in dashed boxes attached to the nodes and edges. The HDM supports instance-based semantics and so each schema object has an **extent** that is the set of values from the data source that the node or edge represents. For example the node $\langle\langle \mathsf{s_{sql}} : \mathsf{bmi} \rangle\rangle$[1] and the edge $\langle\langle \_, \mathsf{s_{sql}} : \mathsf{weight}, \mathsf{s_{sql}} : \mathsf{bmi} \rangle\rangle$ in Figure 2 together represent the bmi column from the weight table in Figure 1. The extent of $\langle\langle \mathsf{s_{sql}} : \mathsf{bmi} \rangle\rangle$ is $\{17, 22, 17\}$. The extent of $\langle\langle \mathsf{s_{sql}} : \mathsf{weight} \rangle\rangle$ is defined as the extent of the primary key of the table and is therefore $\{100, 101, 103\}$. The extent of the edge is $\{\langle 100, 17\rangle, \langle 101, 22\rangle, \langle 103, 17\rangle\}$. Since the extent of $\langle\langle \mathsf{s_{sql}} : \mathsf{weight} \rangle\rangle$ is that of the primary key of the table its values are unique and thus the extent of any edge linked to it cannot include repeated values. This is represented by the unique ($\triangleleft$) constraint from $\langle\langle \mathsf{weight} \rangle\rangle$ to $\langle\langle \_, \mathsf{weight}, \mathsf{bmi} \rangle\rangle$. The fact that every value in the bmi column must have an associated primary key value is represented by the mandatory ($\triangleright$) constraint. A full description of the constraint operators can be found in [2].

Inter Model Schema and Data Integration To merge the HDM graphs in Figures 2 and 4 to form the public schema in Figure 5 we use the BAV data integration method [2]. In BAV, schemas are mapped to each other using a sequence of *bidirectional* schema transformations called a **pathway**. The technique readily supports schema evolution [4] which can be expressed as extensions to existing pathways. This feature makes BAV well suited to P2P data integration where peers may join or leave the network at any time, or may change their set of local schemas, published schemas, or pathways between schemas.

A BAV pathway is created by the repeated application of one of the 5 primitive BAV transformations: **add** and **delete**, **extend** and **contract** and **rename**. Each of these transformations returns a schema that differs from the input schema by a single construct. The **add** and **delete** transformations include a query that defines the extent of the new or removed schema object in terms of the extents of existing schema objects. When it is not possible to define the extent of the new schema object in terms of existing schema objects the **extend** and **contract** transformations are used. This allows us to define transformation pathways between schemas that are not semantically equivalent. This is a particularly useful feature in a heterogeneous P2P environment where schemas may not have the same information capacity. In the example both s_{sql} and s_{xml} are transformed to match s_{public} and then merged using append semantics. We

[1] We use double angle brackets $\langle\langle\rangle\rangle$ to denote the schema objects in a schema

nhs_num

id
100
101
102
103
104

blood

idblood	bsl
100	3
101	3
103	5
104	4

weight

idweight	bmi
100	17
101	22
103	17

blood.idblood ← nhs_num.id

weight.idweight ← nhs_num.id

Fig. 1. SQL database

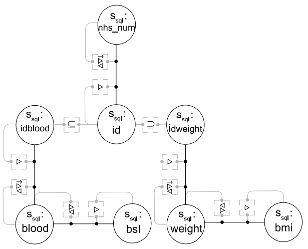

Fig. 2. s_{sql}

```
<results>
    <result id = "102">
        <bsl>4</bsl>
        <bmi>17</bmi>
    </result>
    <result id = "103">
        <bsl>5</bsl>
    </result>
</results>
```

Fig. 3. XML document

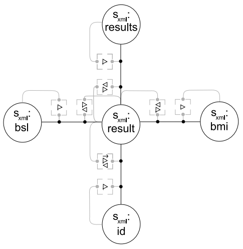

Fig. 4. s_{xml}

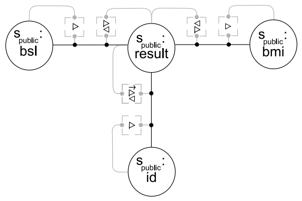

Fig. 5. s_{public}

assume that the mappings between the nodes and edges in the different schemas are provided by a data expert. Rizopoulos [5] has implemented a semi-automatic match and merge algorithm using BAV but a discussion of this beyond the scope of this paper.

3 Schema Metadata Management in a P2P Environment

The idea behind schema metadata management is to abstract from the public schema objects a number of essential attributes which guarantee a good schema search and at the same time requires less bandwidth and storage space than the public schema definition. These attributes form a schema object metadata, or s-o-m, and include the schema object scheme, the schema identifier, the transformation query, and schema object usage. The schema identifier consists of the peer's identifier (e.g. name) and the schema name. The schema object usage is measured by a function over a range of parameters driven by how often a schema object is used by peers in the network.

The s-o-ms are distributed to a schema metadata repository, $\mathcal{SMR}$, which is formed using a P2P routing protocol $\mathcal{R}$. We define $\mathcal{SMR}$ as a tuple: $\langle \mathcal{D}, \mathcal{P}, \mathcal{SO}^{\star}, \overset{\mathsf{p}}{\hookrightarrow}, \overset{\mathsf{o}}{\looparrowright}, \mathcal{R}, \mathcal{U} \rangle$, where $\mathcal{D}$ is the set of peer domains (or groups), $\mathcal{P}$ is the set of peers on the network, $\mathcal{SO}^{\star}$ is the set of a schema object metadata, $\overset{\mathsf{p}}{\hookrightarrow}$ and $\overset{\mathsf{o}}{\looparrowright}$ are peer and schema object metadata mapping functions, and $\mathcal{U}$ is the set of domain values of the usage statistics. A key to the efficiency and robustness of the $\mathcal{SMR}$ is the routing protocol $\mathcal{R}$ which defines the id-mapping functions

$\overset{\mathsf{p}}{\hookrightarrow}$ and $\overset{\mathsf{o}}{\not\hookrightarrow}$. For example, we could use a DHT-based protocol [6] which scales gracefully ($O\,\mathsf{log}(\mathsf{N})$) with the network size and guarantees with high probability a random distribution of the value objects.

Search Schema Searching for public schemas is reducible to searching the $\mathcal{SMR}$ for s-o-ms whose transformation queries match a given criteria. Once the s-o-ms have been found, we can extract from them the schema identifiers and use these to wrap the corresponding public schemas to perform integration. We use a predicate-based query and a quality factor as the search criteria. A predicate-based query, $\mathsf{Q^s}$, is defined as a tuple ⟨SSchemes, SPreds⟩ where $\mathsf{SSchemes}$ is a set of schema object schemes and SPreds is the set of predicates defined over $\mathsf{SSchemes}$. Formally, given a search query $\mathsf{Q^s} = \langle \mathsf{SSchemes}, \mathsf{SPreds} \rangle$ and a quality factor qa^ℓ retrieve the set of schema object metadata $\mathsf{SO^M}$ such that for every $\mathsf{s\text{-}o\text{-}m} \in \mathsf{SO^M}$, $\exists\, \mathsf{sc} \in \mathsf{SSchemes}$ s.t all of the followings are true: **(1)** $\mathsf{match(s\text{-}o\text{-}m.scheme, sc)}$ **(2)** $\forall \rho_{\mathsf{sc}} \in \mathsf{SPreds}$ then $\mathsf{checkSatisfiability}(\mathsf{s\text{-}o\text{-}m.q}, \rho_{\mathsf{sc}})$, **(3)** $\nexists\, \mathsf{s\text{-}o\text{-}m'} \in \mathsf{SSchemes}$ s.t. $\mathsf{s\text{-}o\text{-}m'.schema} = \mathsf{s\text{-}o\text{-}m.schema}$ and $\mathsf{s\text{-}o\text{-}m'}$ satisfies (1) but not (2), and **(4)** $\mathsf{rank(s\text{-}o\text{-}m)} \geq \mathsf{qa}^\ell$. The $\mathsf{checkSatisfiability}$ function is a distributed form of the query answering using views problem [7] but runs faster because we omit the final phase where the satisfied views are combined into the final rewritings. The resulting s-o-ms are ranked based on the quality factor qa^ℓ.

Of particular interest to our schema metadata management is the P-Grid protocol [8] which uses a prefix-based routing strategy that supports the schema object scheme of a s-o-m. To search for schemas in P-Grid, we would program the peers to construct a binary tree of the id space of the s-o-ms and organise themselves along this tree's paths. We then rewrite a search query into a conjunction of scheme-based sub-queries and forward each sub-query to the peer(s) responsible for the scheme of this sub-query. The target peers check satisfiability of the queries and results are returned to the source peer where they are finally combined to determine the resulting s-o-ms.

References

1. Batini, C., Lenzerini, M., Navathe, S.B.: A comparative analysis of methodologies for database schema integration. ACM Comput. Surv. **18**(4) (1986) 323–364
2. Boyd, M., McBrien, P.: Comparing and transforming between data models via an intermediate hypergraph data model. J. Data Semantics IV (2005) 69–109
3. McBrien, P., Poulovassilis, A.: A general formal framework for schema transformation. In: Data and Knowledge Engineering. Volume 28. (1998) 47–71
4. McBrien, P., Poulovassilis, A.: Schema evolution in heterogeneous database architectures, a schema transformation approach. In: CAiSE. (2002) 484–499
5. Rizopoulos, N., McBrien, P.: A general approach to the generation of conceptual model transformations. In: CAiSE. (2005) 326–341
6. Balakrishnan, H., Kaashoek, M.F., Karger, D., Morris, R., Stoica, I.: Looking up data in p2p systems. In: Communications of ACM. (2003)
7. Halevy, A.: Answering queries using views: A survey. VLDB Journal **10**(4) (2001) 270–294
8. Aberer, K.: P-Grid: A self-organizing access structure for P2P information systems. Proc. of CoopIS 2001 **2172** (2001) 179–194