Information Systems 39 (2014) 80–107

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Integrating domain heterogeneous data sources using decomposition aggregation queries

Jian Xu*, Rachel Pottinger

Computer Science Department, The University of British Columbia, 201 2366 Main Mall, Vancouver, Canada V6T1Z4

ARTICLE INFO

Article history:
Received 12 April 2011
Received in revised form
28 November 2012
Accepted 13 June 2013
Recommended by: L. Wong
Available online 19 June 2013

Keywords:
Semantic integration
Aggregation
Query optimization

ABSTRACT

The decomposition aggregation query (DAQ) we introduce in this paper extends semantic integration queries by allowing query translation to create aggregate queries based on the DAQ's novel three role structure. We describe the application of DAQs in integrating domain heterogeneous data sources, the new semantics of DAQ answers and the query translation algorithm called "aggregation rewriting".

A central problem of optimizing DAQ processing requires determining the data sources towards which the DAQ is translated. Our source selection algorithm has cover-finding and partitioning steps which are optimized to 1. lower the processing overhead while speeding up query answering and 2. eliminate duplicates with minimal overhead. We establish connections between source selection optimizations and classic NP-hard optimizations and resolve the optimization problems with efficient solvers. We empirically study both the DAQ query translation and the source selection algorithms using real-world and synthetic data sets; the results show satisfying scalability both in size of aggregations and data sources for the query translation algorithms and the source selection algorithms save a good amount of computational resources.

1. Introduction

Many applications require querying multiple databases with heterogeneous schemas. We refer to all systems that translate queries over databases and thus support querying multiple, heterogeneous, independently maintained databases as *semantic integration systems*. Many semantic integration architectures exist, including data integration (e.g., [26,27]), peer data management systems (PDMSs, e.g., [4,28]), and dataspaces [20].

Existing semantic integration approaches generally focus on cases where the *schemas* are heterogeneous, but the entities in the sources are from the same domain — the data sources are *domain homogeneous*. For example, querying independently maintained bibliography databases is a widely used example for integrating domain homogeneous data sources. Although bibliographic records may have different schemas in different data sources, the schemas represent the same *kind* of objects. Semantic integration must also manage *domain heterogeneity*. For example, the 50+ Amazon public data sets [3] are categorized into 8 domains and the 7144 databases on Freebase [21] belong to 86 domains. Integrating domain heterogeneous schemas is more challenging than integrating domain homogeneous schemas since it requires transforming attributes of objects in one domain to those in another domain. One major difficulty is that entities in one domain may not have direct correspondences in another domain. Although existing semantic integration systems support simple transformations such as concatenating *first name* and *last name* in one schema to form a *full name* in another schema, integrating domain heterogeneous data requires additional support. As shown in the following

* Corresponding author. Tel.: +1 9494078218.
E-mail addresses: shelbyxu@gmail.com (J. Xu), rap@cs.ubc.ca (R. Pottinger).

0306-4379/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.is.2013.06.003

example, motivated by a real-world disaster management project (JIIRP [36]), the new demands require developing new techniques to manage associations between *domain heterogeneous* entities and answer cross-domain queries.

Example 1.1 (*Cell–building heterogeneity*). Consider planning responses to disasters. The planners' domain abstracts infrastructure elements as "cells," which are logical units that perform a single function (e.g., a hospital complex) and the engineers model seismic damage on "buildings." □

Traditional semantic integration systems motivate their work with both domain homogeneous and domain heterogeneous examples. However, the mappings used by existing semantic integration systems typically do not handle domain heterogeneous sources. For example, traditional mappings (e.g., [49,58]) cannot translate queries between the cells and buildings in Example 1.1 since there are no common objects to relate each other. The mappings in [39,40] consider objects from different domains; however, the system still requires common *attributes* to relate data records (e.g., a gene-record and a research paper about the gene are linked by the gene's id). Deeper semantic relationships between objects, e.g., those that require aggregation, as in the following example, are not managed or used for integration.

Example 1.2 shows a simplified scenario of how the relationships between heterogeneous objects in Example 1.1 can be expressed using aggregation:

Example 1.2 (*Aggregation between cells and buildings*). Table 1 shows a snapshot of the cells (in the planners' domain) and building damage assessment (in the engineers' domain). A cell's damage (`Cell.damage`) is estimated by *averaging* the damage to the constituent buildings (`avg (BdnDmg.Damage)`). Monetary loss (`Cell.loss`) is estimated by *summing* the buildings' losses (`sum (BdnDmg.Loss)`).

Lacking systematic integration, the `Cell` table is manually populated by aggregating records in the `BdnDmg` table for each cell (e.g., `C1` and `C2`). This is challenging because those calculating the losses are unfamiliar with seismic damage assessment. Additionally, the laborious process of populating a cell discourages users from defining new cells or updating cell attributes. Our goal is to systematically transform queries on cells into aggregate queries over building seismic assessments (i.e., to automatically compute the "unknowns" in the `Cell` table in Table 1). □

Computing the *damage* and *loss* attributes in Example 1.2 requires translating queries on *Cells* into aggregate queries on *BdnDmg* data. One difficulty is that if the users do not know the data sources beforehand, it is impossible to predetermine the aggregate queries associated with the *Cells*.

This problem motivates the focus of this work: *How can we bridge the gap between domain heterogeneous data sources by automating transforming queries over complex compounds into the components that form them.*

Answering domain heterogeneous aggregate queries is challenging for three reasons: (1) domain heterogeneity prevents users from manually translating queries due to their limited domain knowledge; (2) users may not know (or even desire to know) that their query requires aggregating data from multiple sources and (3) multiple databases with potential duplicates, varying answers, and different subsets of relevant data must be collected from multiple databases. These challenges require supporting fully automatic query translation at the system level.

In addition to our running example, there are many other cases requiring solution to domain heterogeneous aggregate queries. For example:

- Estimating the cost to build a room. The cost of a room is estimated by decomposing the room into its constituent parts (e.g., windows and beams) and then aggregating their costs from providers' databases. Lawrence et al. [42] focused creating and maintaining the mappings necessary to coordinate updates, not how to choose which part of the aggregation came from which source, or what to do with conflicting values.
- Quickly aggregating data in order to detect fraud or to improve performance analysis. This was studied in the context of streams [23], but did not focus on the fact that the data may come from multiple sources (e.g., identify theft can be better traced by combining information from bank accounts, credit cards, medical records, and criminal records). In this case, the exact figures do not matter—it is quickly finding abnormalities in the overall trend that is a problem.

Table 1
The cell and seismic damage schemas.

Domain A: Infrastructure Interdependency. **Relation**: Cell

Cellid	Cellname	Shape	Damage	Loss
C1	Hospital	s1	"*unknown*"	"*unknown*"
C2	Power House	s2	"*unknown*"	"*unknown*"

Domain B: Seismic Assessment. **Relation**: BdnDmg

DmgBid	Name	Intensity	Damage	Loss (dollars)
3	Koerner	VIII	0.4	9,329,501
4	Purdy	VIII	0.35	3,574,677
⋮	⋮	⋮	⋮	⋮
158	Power House	VIII	0.65	545,833
159	Meter Station	VIII	0.4	1,324,292
⋮	⋮	⋮	⋮	⋮