Quete: Ontology-Based Query System for Distributed Sources

Haridimos Kondylakis¹,², Anastasia Analyti¹, and Dimitris Plexousakis¹,²

¹ Institute of Computer Science, FORTH-ICS, Greece
² Department of Computer Science, University of Crete, Greece
{kondylak,analyti,dp}@ics.forth.gr

Abstract. The exponential growth of the web and the extended use of database management systems in widely distributed information systems has brought to the fore the need for seamless interconnection of diverse and large numbers of information sources. Our contribution is a system that provides a flexible approach for integrating and transparently querying multiple data sources, using a reference ontology. Global semantic queries are automatically mapped to queries local to the participating sources. The query system is capable of handling complex join constructs and of choosing the appropriate attributes, relations, and join conditions to preserve user query semantics. Moreover, the query engine exploits information on horizontal, vertical, and hybrid fragmentation of database tables, distributed over the various data sources. This optimization improves system's recall and boosts its effectiveness and performance.

Keywords: Ontology-based data integration, mediation systems, query processing, table fragmentation rules.

1 Introduction

Data Integration is one of the key problems for the development of modern information systems. The exponential growth of the web and the extended use of database management systems has brought to the fore the need for seamless interconnection of diverse and large numbers of information sources. In order to provide uniform access to heterogeneous and autonomous data sources, complex query mechanisms have to be designed and implemented [16]. The design and implementation of a query mechanism is non-trivial because of the heterogeneity of the various components that are going to be queried.

In this paper, we describe an ontology-based mediator system, called Quete, that provides a flexible approach for transparently integrating and querying multiple relational data sources. In particular, it provides full location, language, and schema transparency for users, and integrates dynamically heterogeneous (and possibly overlapping) relational data sources in evolving environments. A common reference ontology is used across integration domains and the data source-to-ontology annotation process, which follows the Local-as-View approach [3], is performed only once per data source.
The motivation for this work was the integration of four database systems, in order to meet the needs of the PrognoChip project [13]. The aim of the project is to identify classification and prognosis molecular markers for breast cancer, through DNA microarray technology. Specifically, our task was to integrate two Clinical Information Systems that store clinical information about patients of two different hospitals and two Genomic Information Systems that store information on DNA microarray experiment settings and results. The objective was to provide a transparent layer that could enhance knowledge extraction and data exchange between these systems. Specifically, this layer should accept ontology-based queries from tools and users, transparently break these queries into local subqueries based on metadata, send the subqueries to the constituent databases, and integrate the returned results.

Our system is an extension of a preliminary and incomplete version of Unity [10, 6], that provides the data source-to-ontology annotation mechanism and a local subquery formation algorithm. In particular, for each relational data source, a local annotator annotates (the interesting to the user) table attributes with paths over the reference ontology, called semantic names. We extended the local subquery formation algorithm that is provided by Unity, such that system’s recall is increased with no sacrifice in precision. Additionally, we implemented the composition of the local subquery. A novel feature of our system is that horizontal, vertical, and hybrid fragmentation rules about underlying schemata can be declared and used, increasing system’s recall and improving performance. In particular, we consider table fragmentation rules, during (i) the formation of the local subqueries, further extending Unity’s algorithm, and (ii) during the formation of the result composition plan. This assures that local subqueries are formed and composed in such a way that final results, presented to the user, are as if there was no table fragmentation. Further, system’s performance is optimized by eliminating local subqueries and avoiding joins in the result composition plan that are certain to return empty results. We want to note that our approach to data integration is by no means restricted to biomedical informatics. On the contrary, it is completely domain independent.

The rest of the paper is organized as follows: Section 2 reviews related work on mediator-based data integration. In Section 3, our architecture is described, providing details about the data source-to-ontology mappings, the processing of the user semantic query, and the incorporation of the table fragmentation rules into the system. In Section 4, preliminary experimental evaluation of Quete is provided. Finally, Section 5 concludes the paper and gives directions for further research.

2 Related Work

A mediator is a system that is responsible for reformulating, at runtime, a user query on a single mediated schema into a composition of subqueries over the local source schemas [7]. To achieve this, a mapping is required that captures the relationship between the local source descriptions and the mediator schema. Specifying this correspondence is a crucial step, as it influences both how difficult query reformulation is and how easily new sources can be added to or removed from the integration system. The two main approaches for establishing the mapping between each source schema and the global schema are the Global-as-View (GAV) and the Local-as-View (LAV).