Chapter 8.17
General Strategy for Querying Web Sources in a Data Federation Environment

Aykut Firat
Northeastern University, USA

Lynn Wu
Massachusetts Institute of Technology, USA

Stuart Madnick
Massachusetts Institute of Technology, USA

ABSTRACT

Modern database management systems are supporting the inclusion and querying of non-relational sources within a data federation environment via wrappers. Wrapper development for Web sources, however, is a convolution of code with extraction and query planning knowledge and becomes a daunting task. We use IBM DB2 federation engine to demonstrate the challenges of incorporating Web sources into a data federation. We, then, present a practical and general strategy for the inclusion and querying of Web sources without requiring any changes in the underlying data federation technology. This strategy separates the code and knowledge in wrapper development by introducing a general-purpose capabilities-aware mini query-planner and a data extraction engine. As a result, Web sources can be included in a data federation system faster, and maintained easier.

INTRODUCTION

Federated databases offer information integration on demand in dynamic environments, where data warehousing approaches are not feasible (Sheth & Larson, 1990; Geer, 2003). In modern relational database management systems, even non-relational sources can be included in a data federation via "wrappers" so that they can be queried as if they

are part of a single large database (Somani, Choy, & Kleewein, 2002; Thiran, Hainaut, Houben, & Benslimane, 2006). Wrappers are mechanisms by which the federated server interacts with non-relational data sources by performing operations such as connecting to a data source and retrieving data from it iteratively.

Retrieving data from Web sources, however, is complicated because data is semistructured and Web sources may have requirements (e.g., they may require forms to be filled before returning data); thus general-purpose wrappers for arbitrary Web pages are not provided in data federation systems. Instead the user needs to implement a custom wrapper for each Web source by coding data extraction patterns and parts of the federated query planning protocol in a low-level programming language such as C. This convolution of code with the data extraction and planning knowledge turns wrapper development into a daunting task, results in code duplication, and slows down the data federation process.

Within the last decade or so, many research projects (Papakonstantinou, Gupta, & Haas, 1998; Levy, Rajaraman, & Ordille, 1996; Li & Chang, 2000; Zadorozhny, Bright, Vidal, Raschid, & Urhan, 2002; Li, 2003; Pentaris & Ioannidis, 2006) offered algorithmic solutions to "query planning with source restrictions." The goal of these studies was to offer an expressive language to specify source restrictions, and let the federated query planner come up with an optimal plan using this knowledge. These approaches do not need any cooperation from the individual data sources other than knowing about their limitations. Had they found their way into commercial systems, they would eliminate part of the code and knowledge convolution problem: the wrapper developer would only need to code the data extraction knowledge and not worry about the query planning aspects. Yet the separation of code and knowledge would still not be satisfactorily achieved in non-cooperative federated query planners. For this study, we have chosen to work with IBM DB2's cooperative federated query planner, which poses more challenges than the non-cooperative ones. Our focus is on improving the usability and maintenance aspects of the wrapper development process without requiring any changes in its underlying data federation technology. We do not offer yet another proposal to rewrite a state-of-the-art distributed query planner (Kossmann, 2000), or create an independent infrastructure for querying Internet data sources (Braumandl et al., 2001; Suciu, 2002), but provide a non-intrusive approach that works with what is available today with minimal effort.

We have tested our prototype implementation with numerous Web sites. A moderate user with no programming experience can include a typical Web site into a data federation in less than an hour. The process often takes much longer when the existing procedural coding approach is used by an experienced programmer. Furthermore, explaining, learning, and tutoring wrapper development becomes much easier, as the task changes from writing and debugging a *program* to specifying and debugging *knowledge*.

In the rest of this paper, we start with a motivational example that illustrates the need for data federation involving Web sources. We then provide some background on data federation with non-relational data sources and describe the current architectural difficulties of incorporating a Web source. Next, we describe our approach to wrapper development, and the algorithms used to perform planning and optimization for Web sources with capability restrictions. We end with an overview of related work and future research issues.

MOTIVATIONAL EXAMPLE

Consider, first, finding the *military expenditure per capita* of countries in the world using the CIA world fact book Web site. This information is scattered inside the world fact book (see Figure 1), and first needs to be located and extracted.