Report on the Workshop on Wrapper Techniques for Legacy Data Systems

Ph. Thiran[1,2], T. Risch[3], C. Costilla[4], J. Henrard[1], Th. Kabisch[5], J. Petrini[3], W-J. van den Heuvel[6], J-L. Hainaut[1]

[1]University of Namur, Belgium
[2]Eindhoven University of Technology, The Netherlands
[3]Uppsala University, Sweden
[4]TU Madrid, Spain
[5]TU Berlin, Germany
[6]Tilburg University, The Netherlands

This report summarizes the presentations and discussions of the first workshop on Wrapper Techniques for Legacy Data Systems which was held in Delft on November 12 2004. This workshop was co-located with the 2004 WCRE conference. This workshop entails to our best knowledge the first in its kind, concentrating on challenging research issues regarding the development of wrappers for legacy data systems.

1. INTRODUCTION

Research and development in legacy data wrapping are of paramount importance given the fact that the combination of distributed Internet technology and wrappers opens up new opportunities to unlock the business value that is stored in legacy systems. In fact, legacy systems hold services that remain useful beyond the means of the technology in which they were originally implemented. Wrappers are used to facilitate the reuse of key portions of the legacy systems for their integration into novel business systems, like web applications.

Grossly speaking, a wrapper can be perceived as a model converter, i.e., a software component that translates data and queries from the interface (model and languages) of a legacy data system to another, abstract interface intended to client applications and users. A critical issue for the development of wrappers is that legacy data systems vary widely in their support for data manipulation and description. Moreover, the data schema of the underlying data source can be unavailable or incomplete. This is typically the case for web sources. In these cases, the data schema should preferably be automatically derived from the actual data that is captured in the (web) resource.

After a short introduction, the day started with a technical session of papers, followed by a plenary discussion about some issues in data wrapping. In the following, a summary of the main points of each of these is presented.

2. TECHNICAL PRESENTATIONS

After a blind review process, four papers were selected for presentation and publication. These workshop proceedings are published on-line by the university press of the Eindhoven University of Technology and are accessible from the workshop website, which will remain open at http://wwwis.win.tue.nl/wrapper04/.

Göldner et al. [1] deal with the maintenance of Web wrappers. They introduce an approach which uses caching of sample data and structures in combination with content recognition to enhance robustness of wrappers and to automate the wrapper maintenance process. Petrini and Risch [2] describe a query translating wrapper of relational databases from RDF-based semantic web queries into SQL to enable access to relational databases from semantic web tools. The particular problem is here efficient dynamic optimization of queries to the wrapper. In their paper, Vila and Costilla [3] present a mediator-wrapper architecture based on an extractor data model placed between each Web data source and its wrapper linked to a specific ontology. Finally, Jean Henrard and al. [4] address the difficult problem of migrating application programs from a legacy data manager, such as a COBOL file system, to a modern DBMS, such as a relational database management system. The approach relies on the concept of *inverse wrappers*; that is, wrappers that simulate the legacy API on top of the new database.

3. DISCUSSION

The workshop was concluded by a plenary discussion in order to provide a definition of wrappers as well as to summarize the state of the art in this research field. In this section, we will briefly sketch the main highlights of the discussions, focusing on wrapper definition and on the evolution of modeling languages.

3.1 Wrapper Definition

The purpose of a wrapper is to make one or several data sources queryable in terms of an agreed-upon query language.

Individual wrappers can be defined for each source, e.g., for an existing on-line legacy system, an existing data production system, etc. To improve code reuse, generic wrappers can be defined for a class of data sources, e.g., for all relational databases providing JDBC drivers and SQL queries, or for HTML interfaces. Generic wrappers permit easy inclusion of new sources into a mediator system. By object-orientation one can further refine this code sharing by providing a hierarchy of generic wrappers, e.g., DB2 wrappers are specializations of general relational database wrappers, etc.

Wrapper can provide several types of query languages for data sources, ranging from sophisticated query languages to low-level data access interfaces. For example, XML or HTML documents on the web, or legacy systems, are not accessed through a query language but rather through a data access interface. The wrapper must then provide enough query processing capabilities. For example, [1] describes a *query enabling* wrapper for web interfaces based on Xpath. By contrast, if the wrapped XML documents are managed by a system providing SQL interfaces, the wrapper is *query translating* and mainly concerned with translating and decomposing queries of the common query language used by the mediator systems into queries of the wrapped data source. Query translating wrappers can be complex, since a data source with a query language such as SQL provides opportunities for advanced optimization choices to split work between the wrapper and the back-end query processor, while a primitive non-query based data source interface provides fewer choices for optimization [2].

Another important aspect of a wrapper/mediator system is how it is accessed from other systems. For example, SQL provides a well established standard for data access and a mediator system providing standardized SQL interfaces, such as IBM DB2 Information Integrator, can be used by many applications. When legacy systems that use an outdated interfaces, such as COBOL file access, are to be left unchanged when upgrading to, say, relational database technology, a component that wraps the relational databases needs to be linked to the legacy application. This can be done either by replacing the native COBOL API with the wrapper (as with Microfocus COBOL), in which case the programs are left unaltered, or by replacing the native I/O statements with wrapper invocations, in which case the programs have to be modified in a simple way that is particularly easy to automate [4].

3.2 Evolution of the Models and Languages

Over the years, several models have been used for describing wrapped data, such as the relational model which has been frequently used. Since XML has become the standard information exchange language through the Web, there has been a shift to using it as the pivot modeling language which the others have to be translated to or from [3]. The recent trends are to use RDF [2]. This shift has been motivated by the need for high level semantic descriptions of web information for use in, e.g, reasoning tools. The original use of XML was for tree-based structuring of individual text documents. For these documents DTDs provide a primitive schema definition language. For more data-oriented use of XML, XML Schema provides a sophisticated XML-based type and structure system.

By contrast, the data model of RDF is based on triples, <subject, predicate, object>, that can annotate any web resource with arbitrary properties referencing other web resources. The RDF model can be seen as a binary relational data model with well-defined semantics based on logic. An RDF database forms a graph of associations between different web resources rather than an XML-markup that describes a single document. The languages built on top of RDF, e.g., RDF-Schema and OWL also are formally based on logic, which makes them more appealing than, e.g., XML Schema, for describing knowledge. This makes XML and its derivatives suited for data integration and high level reasoning.

4. TOPICS FOR FUTURE WORK

The observations drawn at the workshop and our understanding of the whole area of wrapping yield a number of open questions to be discussed in the future, possibly at the next WRAP workshop. Open issues include:

- Both data and web wrappers may be leveraged using Semantic Web technologies (e.g., Thesaurus or Ontology) in order to infuse more context knowledge in the wrapper.
- Regardless of the type of wrapper, an important issue that needs to be resolved is how these wrappers may be developed or generated in an effective manner for different kinds of data sources so that specific wrapper interfaces can be generated for specific data sources. A critical success factor for developing wrappers seems to be to (semi-) automatically acquire deep generic understanding about each kind of mapping between the structure of the data sources and that of the data advertised by the wrapper.
- Another research direction could be directed at making wrappers more robust and enhance the automation degree; this might be based on the integration of statistical approaches/automated learning approaches in order to make the wrapper framework more "intelligent". It seems especially important to overcome difficulties if the structure of data (either provided on the web or in some database/file system) is not entirely known a priori.
- Moreover, the evolution of wrappers, once in place, will most likely become an important issue of future research.
- Lastly, the issue of which functionalities in a Mediator-Wrapper system can be delegated to the wrapper will be an interesting topic. E.g., some approaches rely on a wrapper-supported query planning.

REFERENCES

[1] Ch. Göldner, Th. Kabisch and J. G. Süß, Developing robust wrapper-systems with Content Based Recognition, in *WRAP*, 2004.

[2] J. Petrini and T. Risch, Processing Queries over RDF views of Wrapped Relational Databases, in *WRAP*, 2004.

[3] J. Vila and C. Costilla, Heterogeneous Data Extraction in XML, in *WRAP*, 2004.

[4] J. Henrard, A. Cleve and J.-L. Hainaut, Inverse Wrappers for Legacy Information Systems Migration, in *WRAP*, 2004.