MIWeb: Mediator-based Integration of Web Sources

Susanne Busse and Thomas Kabisch
Technical University of Berlin
Computation and Information Structures (CIS)
sbusse,tkabisch@cs.tu-berlin.de

Abstract

MIWeb realizes a mediator-based integration of heterogeneous metadata sources over the Web. Mediators are well known from database integration. They improve the quality of search results by providing a query language, by automatically coupling sources, and by filtering redundant results.
The MIWeb system contributes mediator architectures in the context of Web sources, metadata standards for integration by using modern web technologies. MIWeb supports the search for learning materials and related publications. It is based on in-house learning object sources, the search engine Google, and the research index CiteSeer. Technical the integration layer uses technologies like HTTP, RDF and the RDF query language RDQL.

1 Introduction

The upcoming flood of data inside the World Wide Web forces the introduction of new technologies that can reduce this information overload. Thereby several problems have to be considered. Web sources are *heterogeneous*, *autonomous*and there exists *no common ontology* as a basis for integration.

Database concepts can help to solve these problems ([FLM98]). One result of the research on database federations and data integration are mediator-based information systems (MBIS) ([Wie97]).

Sources that use a different schema from the mediator are integrated by defining correspondences that describe mappings between the schemas. The explicit specification of correspondences allows the integration and change of sources during the runtime of the system – a prerequisite for the integration of autonomous web sources. The main difficulty is the definition of the mediator schema as it has to cover all aspects of the whole system. Therefore, metadata standards are used as mediator schemas in broader domains like the web ([KS00]), for example the Dublin Core or domain-specific ones. They are easier to manage and allow the integration of all kinds of resources.

Based on works on mediator-based information systems (especially on the encapsulation of semi-structured data sources, the metadata-based integration using correspondences, and on methods for design and evolution ([Les00, Bus02, Kab03])), MIWeb shows how to use mediator concepts for the integration of web sources. The system was developed in a students project in summer 2003.

2 Architecture

The MIWeb system integrates metadata sources describing different types of web documents: the search engine Google [1], the scientific citation index Citeseer , and specific resources for e-learning developed in the NewEconomy (NE) project. MIWeb is based on a mediator architecture. It consists of three main components (see Figure 1): mediator, wrapper, and mapper.

[1] In detail, we use the QEL/RDF wrapper Roodolf ([Roo]).

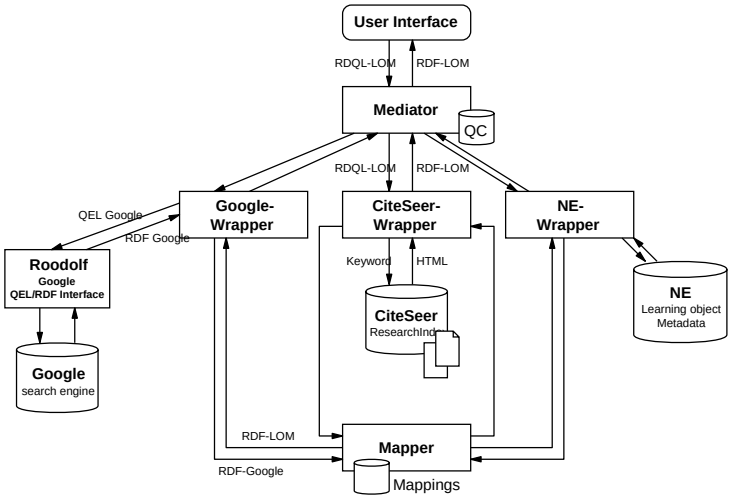

Figure 1: Architecture of the MIWeb system

Mediator In MIWeb metadata is represented as a RDF model, that means the Resource Description Framework RDF is used as the common data model. The mediator schema adheres to the Learning Object Metadata standard (LOM) that is used to describe e-learning resources. Users can query the system with RDQL, a SQL like query language for RDF sources. The mediator is responsible for answering queries against the mediator schema. This includes to generate *plans* (query rewriting), to *execute* these queries by communicating with the wrappers and to *integrate* the results by eliminating redundancies and identifying data conflicts.

Query planning is based on the descriptions of the wrappers' interfaces – the *query capabilities*. Therefore the mediator component also includes a manager for registering, changing, and deleting query capabilities. It is used to dynamically integrate data sources into the system.

Wrapper Wrapping brings the ability to cope with semantic and syntactic heterogeneity and to transform different protocols into the correct mediator standards (technical heterogeneity). The solution of these tasks is done by source-specific wrapper components, the first by a so called *mapper* component, which is used by the wrapper.

The source-specific wrapper functionality depends on the kind of source which should be wrapped. We destinguish structured, semistructured and unstructured sources.

In the MIWeb system most of the wrapping tasks are done with a grammar-based approach, which is suitable for different kinds of sources.

Mapper The mapper is responsible for *resolving semantic and structural heterogeneity* between mediator and wrapper. In our demonstrator it has to transform RDF data of a source-specific schema into LOM-compatible data. The transformation is based on *mappings* manually defined in XSLT. Indicating the query capabilities of the mediator, such mappings are specified explicitely to allow for changes and extensions.

The system also provides a web-based user interface. A search in the MIWeb system encompasses the following steps: the query entered by the user is passed by the user interface to the mediator. The first task then is to plan how to divide the query into sequences (plans) of subqueries to registered sources. When these queries are sent to these sources the wrapper transforms the result into a source-specific representation in RDF. The mapper component translates specific RDF into RDF compliant to the LOM-specification, which is used by the mediator. The mediator component collects all pieces of information delivered by the sources, and integrates them to a result. This is sent back to the user interface that displays it in a human-readable form.

3 Wrapping

Wrappers resolve heterogeneity between the mediator and a data source. We will demonstrate our approach with a exemplarily unstructured source. The research index Citeseer is a bibliographic metadata source for scientific documents. It only provides keyword-based query interfaces. Queries result in an HTML output. The wrapper has to map RDQL queries to these interfaces and the HTML-responses to the desired RDF schema.

Grammar based Wrapping for HTML

We use a grammar based approach for the HTML parsing and interpretation. In terms of a grammar-view the source replies an expression in a "source language" which needs to be interpreted.

In certain cases a type 3 - grammar is sufficient, especially when the text is divided into segments that can be distinguished by grammar tokens. In general only a type 2 - grammar is capable. Our first prototype uses the latter alternative implemented with the compiler compiler SableCC.

Our grammar was defined manually. For future developments automated grammar generation approaches will be taken into account (compare [CMM01]).

Query Tunneling

As mentioned above documents listed in Web Search Engines usually provide no common ontology and a schema-based queriying is uncapable. Thus the query interface is limited to a keyword search. In order to brigde the "semantic gap" to higher-order query languages we propose a Query Tunneling, a two-step algorithm to improve the quality of search results delivered by restricted sources (see figure 2).

At first all selection criteria of the query are extracted and post as keywords to the source. The HTML result will be transformed to RDF by a grammar-based parser described before. In the second step, the original query is executed again against the RDF result. This eliminates data that does not fulfill the given search criteria and improves the quality of the result. As the original query was defined against the LOM schema, the data needs to be transformed before by the mapping component.

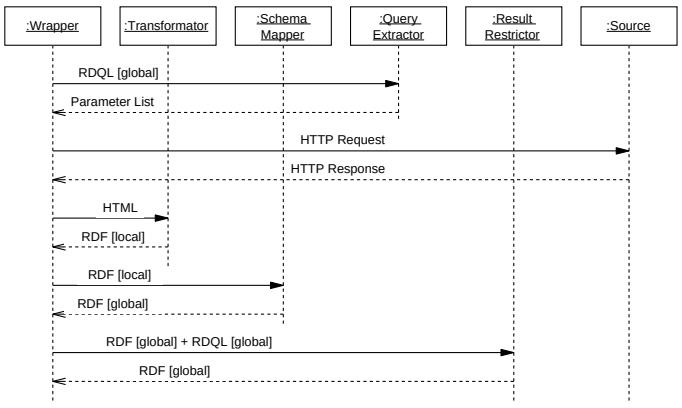

Figure 2: Query Tunneling

4 Mediation

As described in Chapter 2 the mediator is responsible for anwering queries against the global schema - which in our example complies to the Learning Object Metadata standard. The mediator searches for plans of wrapper queries that can answer the global query, executes these queries and integrates the results.

The MIWeb implementation uses Java objects to represent queries and query plans. To handle query results (which are represented within the common data model RDF) we use the Jena API. The result integrator is based on a set of rules for merging the RDF models resulting from the plan executions. Our first prototype merges the metadata belonging to the same learning object (identified by its URL), eliminates duplicate values and merges bags of values building the union. Deeper RDF structures are not considered yet.

Query Planning

For the Query Planning the query capabilities (see for example [GMY99]) of the data sources (wrappers) have to be known. Query capabilities (QC) are defined by parameterized queries ([RSU95]) describing the restrictions of the wrapper's API. The QC shows mandatory parameters and possible result attributes of a wrapper's operation. The semantics are as follows: the operation searches for metadata of learning objects that contain all of the given parameter values and returns all of the result attributes found in the source. The URL of the learning object is always part of the result. For example, the Citeseer wrapper determines the number of citations and the authors of learning objects with a given title:[2]
QC CiteSeer: `title --> author, citations`

In our prototype only such simple kinds of queries occur. Generally there are also operations navigating along several learning objects so that the concept of QC should be extended later on.

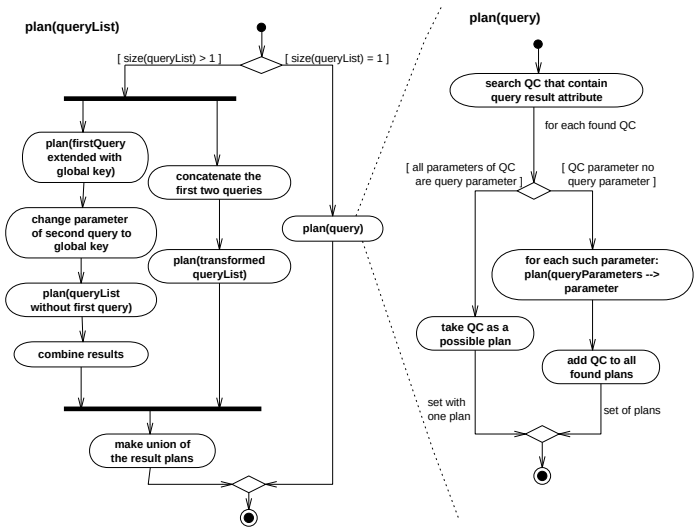

Figure 3: Query Planning Algorithm

The task of query planning is to find all possible sequences (plans) of wrapper queries that determine some attributes being searched within a given query. The algorithm can be divided into three phases:

1. The given RDQL query is translated into a query form similar to query capabilities. Thereby each result attribute is considered without respect to the others. The translation returns a set of queries, one for each result attribute.
2. For each result attribute the planning algorithm is executed. It returns a set of plans of wrapper queries determining results for this attribute. The result of the planning is the union of all of these plans.
3. All plans are checked if they consider all given parameters. In future, we also want to integrate some optimizations in this phase.

[2]The attributes are really named with the property path within the LOM RDF that would be used within a RDQL query. For clarity we use this abbreviated notation.

In contrast to wrapper queries, global queries mostly contain a navigation over several learning objects and their attributes. For example, our mediator is used to enrich the metadata description of referenced learning objects. To allow this kind of queries we don't translate the given RDQL query into one query but into a sequence of queries (in step 1 above). Each element represents one navigation step within the RDF graph. The planning algorithm (step 2 above) is then divided into two parts (see figure 3):

- The outer part identifies navigation paths between resources regarding two possibilities: Either there are wrapper queries covering more than one step (the navigation is related to one resource) or there is a connection between two steps by a global key identifying resources. We use the URL and the title of learning objects as global keys for these connections.
- The inner part searches QCs that can be combined to actualize a query related to one resource. Thereby, the planning algorithm searches automatically for parameters that are required by a wrapper but are not given by the query.

5 Conclusion

The MIWeb system shows how mediator architectures could be used to build high-quality engines searching the web. It improves search engines by allowing semantically richer queries, automatically combining sources and eliminating redundant results.

The system is based on wrappers encapsulating web sources, query capabilities describing the wrappers services and a domain-independent query planning algorithm that determines plans of wrapper queries that give some answers to a given query. This domain-independent solution allows flexible changes of the systems configuration as well as reusing components in other contexts.

Finally we made important experiences with RDF as common data model, the RDF Query language RDQL and the usage of Metadata Standard as mediator schema within a Web context.

References

[Bus02] Susanne Busse. *Modellkorrespondenzen für die kontinuierliche Entwicklung mediatorbasierter Informationssysteme.* PhD thesis, TU Berlin, 2002.

[CMM01] Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In *The VLDB Journal*, pages 109–118, 2001.

[FLM98] D. Florescu, A. Levy, and A. Mendelzon. Database techniques for the world-wide web: A survey. *ACM SIGMOD Record*, 27(3):59–74, 1998.

[GMY99] G. Garcia-Molina and R. Yerneni. Coping with Limited Capabilities of Sources. In *Proc. German Database Conference BTW*, 1999.

[Kab03] T. Kabisch. Grammatikbasiertes semantisches Wrapping für föderierte Informationssysteme. In *Proc. 15. Workshop über Grundlagen von Datenbanken*, 2003.

[KS00] V. Kashyap and A. Sheth. *Information Brokering Across Heterogeneous Digital Data: A Metadata-Based Approach.* Kluwer, 2000.

[Les00] Ulf Leser. *Query Planning in Mediator Based Information Systems.* PhD thesis, TU Berlin, Fachbereich Informatik, 2000.

[Roo] RooDolF 2.0. http://nutria.cs.tu-berlin.de:8080/roodolf2/index.html.

[RSU95] A. Rajaraman, Y. Sagiv, and J. Ullman. Answering Queries Using Templates with Binding Patterns. In *Proc. of the 14th ACM PODS*, pages 105–112, May 1995.

[Wie97] Gio Wiederhold. Mediators in the Architecture of Future Information Systems. In Michael N. Huhns and Munindar P. Singh, editors, *Readings in Agents*, pages 185 – 196. Morgan Kaufmann, San Francisco, CA, USA, 1997.