

Federation Services for Heterogeneous Digital Libraries Accessing Cooperative and Non-cooperative Sources

Martin Endig, Michael Höding, Gunter Saake, Kai-Uwe Sattler, Eike Schallehn
Department of Computer Science, University of Magdeburg
P.O. Box 4120, D-39016 Magdeburg, Germany
{endig|hoeding|saake|kus|schallehn}@iti.cs.uni-magdeburg.de

Abstract

Today, bibliographical information is kept in a variety of digital libraries available on the Internet. The integration of bibliographical data is considered as one of the most important tasks in the area of digital library community. The available sources of data vary widely in terms of data representation and access interfaces. To overcome this heterogeneity during the last years attempts were made to apply methods developed for information system integration. In this paper we describe our approach of a federation service for digital libraries using the loosely coupled federated system FRAQL that offers a variety of conflict resolution mechanisms. Furthermore, we present different kinds of adapters for accessing cooperative and non-cooperative sources of bibliographical information. In order to integrate systems with limited query capabilities we introduce concepts for source descriptions.

1 Motivation

The integration of existing digital libraries and electronic catalogues of publications is considered as one of the main tasks in the digital library community [22]. Publishers, libraries and computer scientists engage in developing digital libraries and electronic catalogues in order to make the publications themselves (digital contents) and their accompanying meta-data (bibliographical data about the publications) available. In different libraries and catalogues various services based on the provided information are offered for certain area of research, for instance the ACM and IEEE digital libraries providing meta-data and full text for computer science in general and the DBLP (*Computer Science Bibliography*) service provide meta-data for the field of database research. Generally, the provided services differ in their quality, for instance full text versus bibliographical data, quantity and the associated cost model.

Today, the integration of meta-data from the available heterogeneous sources and the identification of objects in this global scope are major problems for the integration of digital libraries. For the latter problem current developments in the WWW technology and probabilistic approaches for identifying same objects seem to be promising. New techniques to tackle the problem of meta-data integration were developed in the area of integration of information systems, such as federated database systems and mediator based architectures. The provision of a single point of access to bibliographical meta-data for the users would allow to look up publications more efficiently. Furthermore, the integration makes it possible to derive information about implicit relationships among data spread over various sources, for instance citation linking information.

For the integration of bibliographical data various projects were started, and newly developed methods and technologies from the research area of federated databases and mediator systems were applied successfully. But some problems remained unsolved, because of the complexity resulting from the high level of heterogeneity regarding data representation, contents and interfaces. All approaches have a similar multi-tier architecture in common that provides adapters and/or wrappers for certain classes of sources to deal with interface heterogeneity and, depending on the approach, various issues of data representation. Due to the interface heterogeneity and varying levels of cooperativeness of data providers all of these approaches have their limitations.

In this paper our approach for integrating heterogeneous digital libraries based on federation services will be presented, whereby both cooperative and non-cooperative sources are taken into consideration. The remainder of this paper is structured as follows. In section 2 we give a short overview of related work. In section 3 we introduce an exemplary scenario for the integration of digital libraries in heterogeneous environ-

ments. The major parts of the paper are sections 4 and 5, describing our approach for the integration of bibliographical data and for the handling of cooperative and non-cooperative sources. Section 6 concludes the paper and gives an outlook.

2 Related Work

The integration of various sources for bibliographical information is one of the major tasks in the development of digital libraries. Concepts from the area of information system integration are adopted to this usage scenario. The currently most applicable solutions are based on loosely coupled federated database systems in connection with a multi-database language and mediator-based systems. A federated database system is a distributed system consisting of different heterogeneous database systems or local file clusters [13]. Good overviews are given in [8] and [4]. Query languages supporting the integration of heterogeneous sources are for instance multi-database languages like MSQL [11] and SchemaSQL [14]. Examples of system implementations are federated database systems like IRO-DB [10], Pegasus [1] or IBM DataJoiner [24] as well as mediator-based systems like TSIMMIS [9], Garlic [5] or Information Manifold [15]. Some of the above mentioned systems were used in or developed in conjunction with integrated digital library systems, namely TSIMMIS, Garlic and Information Manifold. The TSIMMIS successor MIX [3] employs XML for data exchange between the mediator and the wrapper.

Another focus is on the integration of non-cooperative providers, especially those offering Web interfaces. The Unicats project [23] integrates digital libraries offering public HTTP interfaces. Other approaches include WebJDBC [20] and techniques for wrapper generation introduced in [2]. TSIMMIS, Information Manifold and Garlic focus on this approach to some extent, too.

We consider Z39.50 sources as especially important in this application scenario. Harp [17] extends an existing DBMS to include Z39.50 sources by using user defined functions. ZBroker [18] is a pure broker for Z39.50 query processing, offering a global Z39.50 interface for sources of the same type.

The description of query capabilities for sources and its usage in query processing was for instance discussed related to the TSIMMIS [16], Garlic [12] and Information Manifold [15] systems.

3 Example Scenario

Today, a great number of different digital libraries exists on the Internet. Such a library can contain data about *one* research field, *one* application scenario or provider specific data. Moreover, in some digital libraries citation information from different areas is available. If users need cited publications from various areas, then they must know which citations are contained in which digital libraries and how to access these libraries. In order to solve this problem, we want to provide an integrated digital library based on existing libraries. Consequently, for the users a uniform interface to bibliographical data from different areas will be supported.

In general, for the integration of bibliographical data in heterogenous environments two important aspects have to be considered. On the one hand, the access to a data source depends on the local query interface and the local query capabilities of the source. Therefore, for the integration of digital libraries a general description of the query capabilities is necessary. On the other hand, in each local data source a specific data format is used, that has to be transformed into a common format. For this transformation process appropriate mapping information are needed.

Presently, in the WWW different providers of bibliographical meta-data offer proprietary query interfaces. In order to enable the integration of data, it is necessary to know, what kind of query capabilities the data sources support. For instance, in some sources only a separate access to authors, titles or keywords is permitted. In other sources very complex query interfaces for querying various combinations of attributes are provided. The combination of attributes is generally based on logic operations. Normally, some providers support different kinds of query interfaces for different kinds of users, too. Two special providers of bibliographical data are the *LINK Information Service* from the Springer publishing house and the *Computer Science Bibliography* (DBLP) from the University of Trier. The providers support both simple and complex query interfaces. While the access to DBLP is very restricted (author, title or keyword search), Springer Link also includes the access to abstracts and full texts. These simple examples show, that the description of source capabilities is necessary for the integration of digital libraries in a heterogenous environment.

Apart from the description of source capabilities the handling of local data formats is another important aspect. While concepts for the integration of structured data in database systems have proven successful during the last years, the integration of autonomous sources via public HTML interfaces is more problematic, due

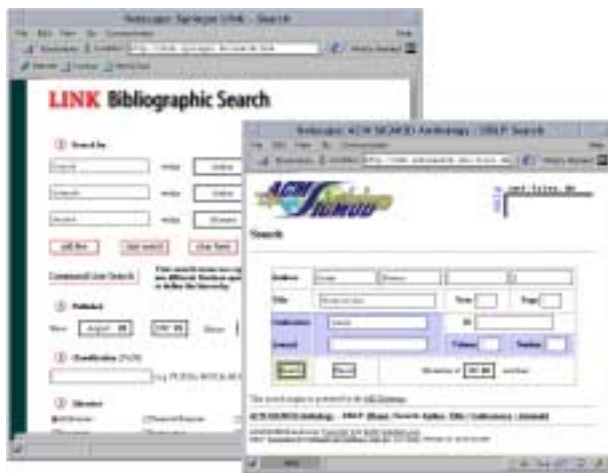


Figure 1: Examples for Proprietary WWW Query Interfaces for Digital Libraries

to various technical problems. The stability in changing environments and the extraction of data will probably remain error-prone. More stable and more efficient solutions are conceivable if we assume a certain level of cooperativeness from the providers of bibliographical data.

4 Federation Services

The objectives of federation services are to provide a uniform interface hiding the special features and restrictions of the individual sources, and to supply an integrated view on the data. Queries on the global views have to be decomposed, rewritten, and possibly translated in order to be processible by the sources. Finally, the results of the sub-queries have to be combined which means resolving conflicts caused by different representations of data.

In the following sections we present the architecture of our federation service for digital libraries. Furthermore, we discuss the main issues of query processing and rewriting as well as result set integration.

4.1 Architecture

In principle, the architecture of our federation service comprises three layers: on the *data source layer* all bibliographical data are provided preserving the autonomy of data sources, on the *adapter layer* special adapters for different kinds of sources implement a uniform access interface and transform the source-specific data model into the global model, and the *federation layer* is responsible for query rewriting and processing as well

as data integration. In this section we focus the discussion on the federation and adapter layers.

Federation Layer The federation layer provides services for defining and querying integrated views on bibliographical data from different sources. These services are implemented by using the FRAQL language and query processor [19]. FRAQL is a query language for object-relational database federations. It extends SQL by features for defining federations, accessing meta-data in queries, restructuring query results, and resolving integration conflicts. Furthermore, it is extensible by user-defined data types as well as functions and it supports dynamic integration of new sources.

In FRAQL a federation is a set of databases consisting of relations. A database can be provided by a full-featured DBMS or even by a Web source encapsulated by a wrapper. FRAQL is based on a simple object-relational data model: it supports the definition of object types and object tables derived from types. *Object types* describe the structure of objects as sets of attributes and their domains. Types can be organized in a specialization hierarchy. *Object tables* represent global virtual relations of the federation. Here we distinguish between import and integration relations. An *import relation* is a projection of a local relation of a data source, where for each global attribute a mapping onto a local attribute can be defined. This mapping can be described

- as simple renaming,
- as value conversion by using a user-defined conversion function
- or with the help of an explicit mapping table in the form $g_attr \text{ is } @tbl (l_attr, src, dest, default)$. This means that the database table tbl is used for mapping the values from the local attribute l_attr . The value of the global attribute g_attr is obtained by looking for the value of attribute l_attr in column src and retrieving the corresponding value of column $dest$. The field $default$ denotes a default value, either as literal or as local attribute, which is assigned to the global attribute, if the value of l_attr is not found in the table.

An *integration relation* is a view on other global relations combined by using operators like union, θ -join and outer join, which are extended by reconciliation functions for conflict resolution [19]. In addition, the standard SQL operations selection and projection are provided, too.

Adapter Layer The adapter layer consists of special components called adapters providing an uniform access to the sources, hiding the differences of the query interfaces and data models. In principle, we can distinguish between two kinds of adapters: *database adapters* for accessing “real” database systems and *Web or file adapters* for integrating Web-based sources and structured files. Database adapters can simply pass through the rewritten query. So, they are responsible mainly for hiding the proprietary DBMS interfaces. Database adapters are also used in our system for accessing global data required for integration, e.g., meta-data and mapping tables.

Web or file adapters provide access to structured data from local files or remote servers. In general two kinds of sources are available: cooperative sources conforming to a predefined query interface and exchange format like a XML-DTD as well as non-cooperative sources providing only a simple query interface and HTML or structured text as a result. Both kinds of adapters share some common components (Fig. 2(b)): a query translator transforming a SQL query into an appropriate HTTP request or file access, a component for accessing meta-data, i.e., information about the structure of the provided data, and a result translator for transforming the query results into the global data model. In addition, the adapter for cooperative sources contains an XSLT processor for transforming a result document of a provider-specific DTD to the adapter-specific DTD and a XML parser for translating the XML data to the internal data format.

In Fig. 2(a) the overall architecture of the federation service is shown. The main part is formed by the FRAQL query processor consisting of the following components: the query parser, the decomposer and the global optimizer, the query evaluator, the Java VM for evaluating user-defined functions, and the catalog for storing meta-data. The adapter layer contains the management component as well as the individual adapters. The interfaces to the adapters and to the query processor itself are implemented using CORBA. Therefore, adapters can be plugged into the system at runtime.

4.2 Query Processing and Reconciliation

Due to the fact that our federation service is based on the FRAQL query language, which extends SQL by features for defining integration views, ordinary SQL queries can be used in order to access the integrated data. In principle, a global query is processed as follows [25]: first the query is transformed by the parser into a query tree. Second the optimizer performs some simple, heuristic-based reordering operations in order to

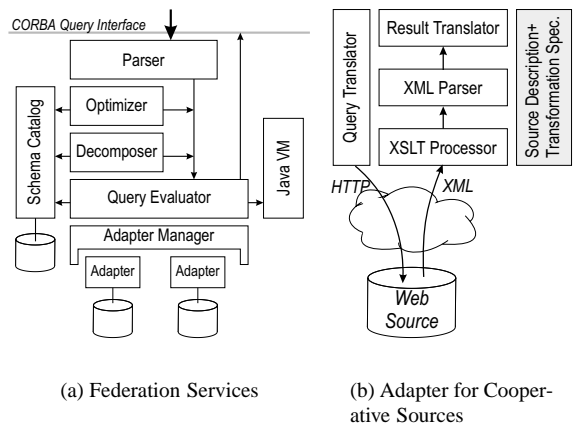


Figure 2: Architecture

minimize the global processing effort. Next, the query is decomposed into sub-queries according to the definition of global relations. These sub-queries are sent to the adapters and evaluated by the data source. Finally, the results are combined corresponding to the query tree.

However, we have to consider two kinds of sources. If the source system is a fully-fledged relational database system we can simply pass through the rewritten sub-query. The second class of sources consists of systems like Internet databases offering Web interfaces. Usually these sources support only constant selections with predefined comparison operators. Conforming predicates can be combined conjunctively. As an example consider two sources exporting the relations *Books* and *BookStore*, both conforming to a global type *Publication*. Source₁ exports relation *BookStore* and supports equality matches on *author* or *title* or a combination of both. Source₂ exports relation *Books* and supports equality match on *title*. These source descriptions are specified in FRAQL as query constraints via a special *alter table* statement for each relation of a source with limited query capabilities. For instance, the constraints for import table *BookStore* are specified as follows:

```
alter table BookStore
set query constraints (
  predicates (authors, =), (title, =),
  combinations (authors), (title),
                (authors, title));
```

Based on this information queries accessing limited sources are rewritten in a way, that the specified query constraints are fulfilled. Assuming a global relation *Publications* as union on *BookStore* and *Books* the rewriting of the following query results in the query tree shown in Figure 3.

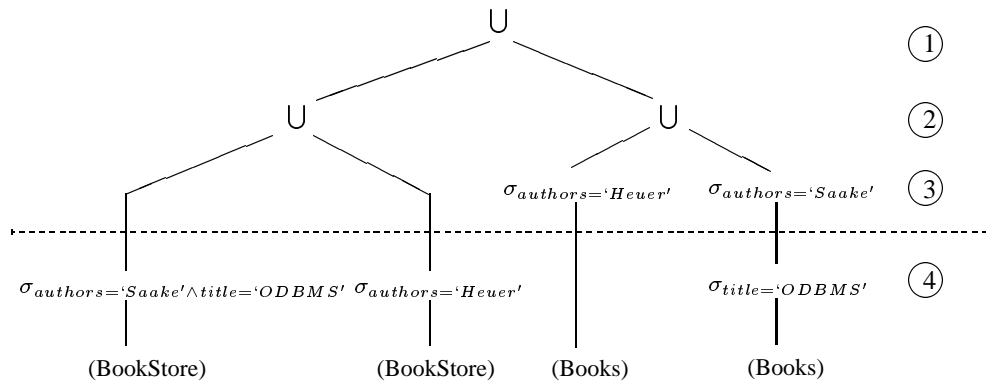


Figure 3: Re-written query example for source systems

```

select *
from Publications
where title = 'ODBMS' and
       authors = 'Saake' or
       authors = 'Heuer';

```

During integration of the results from the sub-queries instance-level conflicts can occur, which are caused by different representations of data values corresponding to the same real-world fact. These conflicts are resolvable by using the extended union and join operators provided by FRAQL, which allow to apply reconciliation functions. A reconciliation function is a user-defined function, which is called for each pair of tuples fulfilling the comparison condition. The affected tuples are passed as arguments to the function and the result is inserted into the global relation. In this way, the value of the global attribute can be computed from the possibly conflicting values of the corresponding attributes. This mechanism is described in [19].

5 Adapter Development

As mentioned in section 4 adapters (often called wrappers) are essential components to cover heterogeneity of information sources. As discussed before one has to distinguish cooperative from non-cooperative information sources. In subsection 5.1 an adapter for cooperative digital libraries is presented. The adapter uses XSLT for transforming source specific XML documents to common XML documents according to the DTD processible by the federation layer. Unfortunately many valuable information systems present information in own formats and via various interfaces. For these non-cooperative sources specific adapters have to be developed. This aspect will be discussed in subsection 5.2.

5.1 Cooperative Providers

While there are suitable solutions for non-cooperative providers offering for instance only access to their data via Web interfaces, more efficient and less error-prone solutions are possible, if we assume a certain level of cooperativeness. Therefore, we consider the following aspects and address them in our approach:

Minimal provider-side resource consumption:

Our approach takes the possibly limited query capabilities into consideration. Query processing is split between source and integration layer according to a source description. For cooperative sources the results are provided by the source as XML conforming to a source-specific DTD. All further transformation is performed in a generic, configurable adapter that is part of the global system.

Minimal provider-side implementation efforts: On the one hand, we simply rely on query capabilities of the source system that can easily be wrapped by the source-specific part of the adapter. The design can be supported by tools using the source description. This is one of our current research interests. On the other hand, the provider supplies XML results conforming to their specific DTD.

Efficient transfer: One critical aspect is the size of results that have to be transferred over possibly slow network connections. This is done by moving sub-queries with high selectivity to the source where possible. XML is suitable as a transfer format, because it is an emerging standard for data transfer on the Internet and can provide type checking, compaction etc.

Efficient access for global applications: By offering an object-relational interface to global applications

we rely on applied and powerful standards. The integration layer offers additional query capabilities allowing the user to post standard SQL queries. Furthermore, it is flexible in terms of data to be imported and ways to integrate this data. The global schema can be modified and additional information, for instance about citations, can be added to the schema.

The approach for this kind of generic adapter is based on the assumption that all relevant results of requests are received as provider specific XML documents. During the exchange of bibliographical data different kind of publications with different kind of attributes in different formats are considered. Therefore, for the implementation of the adapter, the introduction of a well-defined XML format *and* an appropriate transformation is needed. Due to the used concepts and methods on the federation layer, the general format is derived from its object-relational model.

In order to use the general format the introduction of a mapping between the XML conforming to a provider specific DTD and the adapter specific XML format is required. For this purpose we propose the application of XSLT [6] in the adapter. XSLT is a style sheet language for transforming XML documents to other XML documents. It is used for transforming provider specific XML data to XML data suitable for processing by the adapter. Hence, not only the special XML data but also an appropriate XSLT file is required as an input for the adapter. The XSLT file describes significant structures of source documents and their mapping to a result document.

For instance, the execution of a simple query on a data source can produce the following result conforming to the provider-specific DTD:

```
select *
from Publications
where title like '%ODBMS%';
```

The retrieved result is:

```
<ROWSET>
  <ROW num="1">
    <ID>1</ID>
    <TITLE>ODBMS</TITLE>
    <AUTHORS>Heuer</AUTHORS>
  </ROW>
</ROWSET>
```

For this special document type an appropriate XSLT file is required in order to transform it into an adapter-specific XML document. The XSLT processor searches the input document for special patterns and applies the substitutions described in the templates. The templates

are contained in the XSLT file and describe the target document type. Using an XSLT file the following result is created in the adapter:

```
<COLLECTION>
  <RESULTSET tablename="PUBLICATIONS">
    <RECORD>
      <FIELDVALUE name="ID">
        1
      </FIELDVALUE>
      <FIELDVALUE name="TITLE">
        ODBMS
      </FIELDVALUE>
      <FIELDVALUE name="AUTHORS">
        Heuer
      </FIELDVALUE>
    </RECORD>
  </RESULTSET>
</COLLECTION>
```

This document can be processed in the generic adapter, which passes it through to the integration layer for further processing.

5.2 Non-cooperative Providers

Nowadays only a limited (but rising) number of providers of digital libraries is using XML techniques for data exchange and can therefore be called “cooperative”. On the other hand numerous providers offer valuable publications or bibliographical information by common Web techniques using HTML and CGI Interfaces. According to the task of result processing we discuss an adapter, transforming specific HTML result files into the general XML format used by the federation layer.

Processing of files commonly should be done by the use of grammars and parser generators. As argued in [7] this approach is not suitable for Web databases, because Web pages mostly contain a huge part of unstructured, instable, and (from the data extraction perspective) less important data. Instead the grammatical approach an *operational approach* is more suitable. In [20] we present WebJDBC adapters, using an operational description of the result file for result processing based on extraction rules, i.e. patterns consisting of prefixes, infixes, and postfixes for tuples and attribute values. Obviously this kind of source description is not limited to HTML table structures. Prototype implementations written in Java/Phyton and Perl are in use for the digital library project. Tool support for the generation of extraction rules is another focus of the project.

Today, a great number of sources offer bibliographical data via the standard protocol Z39.50. Though the intention behind the development of the protocol was

more on Information Retrieval in general it quickly was adopted to electronic catalogues and digital libraries and can be considered as an application specific interface standard. The protocol itself provides facilities and services on various levels, our focus being on searching meta-data and the presentation of query results.

The *Search Facility* provides boolean search by integrating concepts known from database management systems with those more common in Information Retrieval. As our approach is based on an object-relational integration layer and our goal is to provide structured access to the data, we concentrate on the part of the Search Facility offering database-like features. Those can easily be integrated using an Z39.50 adapter. The considered query facilities consist of simple constant selections with limited sets of queryable attributes, supported comparison operators, and possible combinations based on logical operations. These query capabilities can be described applying the concepts introduced in Section 4.2. Based on this description a global query can be re-written by the integration service to contain only those predicates processible by the source. The re-written query can then be translated to a Z39.50 search request.

The *Presentation Service* allows for negotiating the format of the returned records, but we currently concentrate on records returned in the common MARC format and its various derivatives. Our architectures can easily integrate modules that can cope with other formats as well. A well-known problem is that of finding a mapping between the records existing in a bibliographical format like MARC and the relational or object-relational interface. A bibliographical collection can be interpreted as a universal relation in no normal form. In the adapter multi-valued attributes can be handled applying object-relational extensions offered by the integration service. As an alternative the adapter can also normalize the retrieved data and actually export a more complex schema.

Another problem is the wealth of information that can be expressed using the bibliographical formats. As we use a top down approach for schema integration, this is not a major concern for us. We will only retrieve and extract data that is required by the global application, and by concentrating on a core of relevant fields we will also avoid a number known problems with less frequently used fields.

The concept of server-side result sets and the retrieval of partial results in Z39.50 greatly integrates with the FRAQL and the general adapter architecture. The global query processing takes advantage of pipelining wherever possible. This way processing a result from a Z39.50 source becomes more efficient than those of

sources that require a full fetch.

6 Conclusion

We presented our approach for the integration of bibliographical data that employs the FRAQL federation service for loosely-coupled integrated systems. Furthermore, we introduced concepts for adapters that can be used on a very common class of source systems and allow a seamless integration of cooperative and non-cooperative providers. An adapter for cooperative providers is based on the standards XML for data transfer and XSLT for data restructuring, which are suitable in a highly distributed scenario and keep the transformation process comprehensible. Apart from the XSLT mapping information a query capability description for sources is part of the adapter configuration. Non-cooperative providers are integrated via public HTTP or Z39.50 interfaces.

In summary, our approach represents a suitable solution for the integration of a variety of different sources. As a first example as well as for evaluation purposes we currently implement a simple solution for *citation linking*. Usually citations are used within scientific publications in order to link the contents to related work or publications containing basic or more detailed descriptions of mentioned concepts. This is still very problematic, because a system that only covers a certain area or amount of publications cannot guarantee a meaningful resolution of such a citation. This problem is known as citation or reference linking. It is discussed in more detail in [21].

For further research we investigate tool support for implementing provider-side components and for specifying the XSLT mapping. Another focus is on adding efficient mechanisms for the detection of identical objects during result set integration.

References

- [1] R. Ahmed, P. De Smedt, W. Du, W. Kent, M.A. Ketabchi, W. Litwin, A. Rafii, and M.-C. Shan. The Pegasus Heterogeneous Multidatabase System. *IEEE Computer*, 24(12):19–27, December 1991.
- [2] N. Ashish and C. Knoblock. Wrapper Generation for Semi-structured Internet Sources. In *Workshop on Management of Semistructured Data*. Tucson, Arizona, 1997.
- [3] C. Baru, A. Gupta, B. Ludäscher, R. Marciano, Y. Papakonstantinou, P. Velikhov, and V. Chu. XML-based Information Mediation with MIX. In A. Delis, C. Faloutsos, and S. Ghandeharizadeh, editors, *Proceedings of the ACM SIGMOD International Confer-*

- ence on Management of Data (SIGMod-99), SIGMOD Record, Vol. 28,2, pages 597–599. ACM Press, New York, June 1–3 1999.
- [4] S. Busse, R.-D. Kutsche, U. Leser, and H. Weber. Federated Information Systems: Concepts, Terminology and Architectures. Technical Report Forschungsberichte des Fachbereichs Informatik 99-9, Technische Universität Berlin, 1999.
- [5] M.J. Carey, L.M. Haas, P.M. Schwarz, M. Arya, W.F. Cody, R. Fagin, M. Flickner, A. Luniewski, W. Niblack, D. Petkovic, J. Thomas II, J.H. Williams, and E.L. Wimmers. Towards Heterogeneous Multimedia Information Systems: The Garlic Approach. In Omran A. Bukhres, M. Tamer Özsu, and Ming-Chien Shan, editors, *Proc. RIDE-DOM '95, 5th Int. Workshop on Research Issues in Data Engineering - Distributed Object Management, Taipei, Taiwan*, pages 124–131, IEEE-CS, 1995.
- [6] J. Clark. XSL Transformations (XSLT) – Version 1.0. Technical report, World Wide Web Consortium, November 1999.
- [7] V. Crescenzi and G. Mecca. Grammars Have Exceptions. *Information Systems*, 23(8):539–565, 1998.
- [8] R. Domenig and K. R. Dittrich. An Overview and Classification of Mediated Query Systems. *SIGMOD Record*, 28(3), 1999.
- [9] H. Garcia-Molina, Y. Papanikolaou, D. Quass, A. Rajaraman, Y. Sagiv, J. D. Ullman, V. Vassalos, and J. Widom. The TSIMMIS Approach to Mediation: Data Models and Languages. *Journal of Intelligent Information Systems*, 8(2):117–132, March/April 1997.
- [10] G. Gardarin, S. Gannouni, B. Finance, P. Fankhauser, W. Klas, D. Pastre, R. Legoff, and A. Ramfos. IRO-DB — A Distributed System Federating Object and Relational Databases. In O. A. Bukhres and A. K. Elmagarmid, editors, *Object-Oriented Multidatabase Systems — A Solution for Advanced Applications*, chapter 20, pages 684–712, Prentice Hall, Eaglewood Cliffs, NJ, 1996.
- [11] J. Grant, W. Litwin, N. Roussopoulos, and T. Sellis. Query Languages for Relational Multidatabases. *VLDB Journal*, 2(2):153–171, 1993.
- [12] L. M. Haas, D. Kossman, E. L. Wimmers, and J. Yang. Optimizing Queries Across Diverse Data Sources. In *VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases*, pages 276–285, 1997.
- [13] D. Heimbigner and D. McLeod. A Federated Architecture for Information Management. *ACM Transactions on Office Information Systems*, 3(3):253–278, July 1985.
- [14] L.V.S. Lakshmanan, F. Sadri, and I.N. Subramanian. SchemaSQL - A Language for Interoperability in Relational Multi-Database Systems. In T. M. Vijayaraman, A.P. Buchmann, C. Mohan, and N.L. Sarda, editors, *VLDB'96, Proc. of 22th Int. Conf. on Very Large Data Bases, 1996, Mumbai (Bombay), India*, pages 239–250, Morgan Kaufmann, 1996.
- [15] A.Y. Levy, A. Rajaraman, and J.J. Ordille. Querying Heterogeneous Information Sources Using Source Descriptions. In T.M. Vijayaraman, A.P. Buchmann, C. Mohan, and N.L. Sarda, editors, *VLDB'96, Proc. of 22th Int. Conf. on Very Large Data Bases, 1996, Mumbai (Bombay), India*, pages 251–262, Morgan Kaufmann, 1996.
- [16] C. Li, R. Yerneni, V. Vassalos, H. Garcia-Molina, Y. Papanikolaou, J. Ullman, and M. Valveti. Capability Based Mediation in TSIMMIS. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 27(2), 1998.
- [17] E. Lim and Y. Lu. Harp: a distributed query system for legacy public libraries and structured databases. *ACM Transactions on Information Systems*, 17(3), July 1999.
- [18] Y. Lin, J. Xu, E. Lim, and W. Ng. ZBroker: A query routing broker for Z39.50 databases. In *Proceedings of the 8th International Conference on Information Knowledge (CIKM-99)*, pages 202–209. ACM Press, New York, November 2–6 2000.
- [19] K. Sattler, S. Conrad, and G. Saake. Adding Conflict Resolution Features to a Query Language for Database Federations. In M. Roantree, W. Hasselbring, and S. Conrad, editors, *Proc. 3rd Int. Workshop on Engineering Federated Information Systems, EFIS'00, Dublin, Ireland, June*, pages 41–52. Akadem. Verlagsgesellschaft, Berlin, 2000.
- [20] K. Sattler and M. Höding. Adapter Generation for Extraction and Querying Data from Web Sources. In *Proc. of 2nd ACM SIGMOD Workshop WebDB'99*, 1999. <http://www-rocq.inria.fr/~cluet/webdb99.html>.
- [21] E. Schallehn, M. Endig, and K.-U. Sattler. Citation Linking in Federated Digital Libraries. In *Proceeding of the 3rd International Workshop on Engineering Federated Information Systems, EFIS, 2000. to appear*.
- [22] B. Schatz and H. Chen. Digital Libraries: Technological Advances and Social Impacts. *Computer*, 32(2):45–50, February 1999.
- [23] B. Schmitt and A. Schmidt. METALICA: An Enhanced Meta Search Engine for Literature Catalogs. In *Proceedings of the 2nd Asian Digital Library Conference (ADL'99)*. Taipei, November 1999.
- [24] S. Venkataraman and T. Zhang. Heterogeneous Database Query Optimization in DB2 Universal Data-Joiner. In A. Gupta, O. Shmueli, and J. Widom, editors, *VLDB'98, Proc. of 24rd Int. Conf. on Very Large Data Bases, 1998, New York City, New York, USA*, pages 685–689, Morgan Kaufmann, 1998.
- [25] C. T. Yu and W. Meng. *Principles of Database Query Processing for Advanced Applications*. Morgan Kaufmann Publishers, San Francisco, CA, 1998.