ISSN 2229-5518

Web Information Integration Using Schema Matching

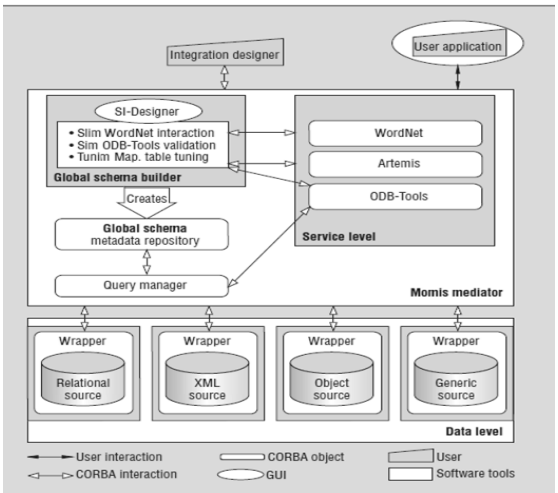
J.Sharmila, Dr. A.Subramani

Abstract : The Web is based on a browsing paradigm that makes it difficult to retrieve and integrate data from multiple sites. Today, the only way to do this is to build specialized applications, which are time-consuming to develop and difficult to maintain. We have addressed this problem by creating the technology and tools for rapidly constructing information agents that extract, query, and integrate data from web sources. Our approach is based on a uniform representation that makes it simple and efficient to integrate multiple sources. Instead of building specialized algorithms for handling web sources, we have developed methods for mapping web sources into this uniform representation. This approach builds on work from knowledge representation, databases, machine learning and automated planning.

Keywords: Hybrid, Interface, Metadata, Ontology, Query, Schema matching, Web Integration

1. INTRODUCTION

The amount of data accessible via the Web and intranets is staggeringly large and growing rapidly. However, the We does not support many Applications by hand. These applications are time-consuming and costly to build, and difficult to maintain.



2. WEB INTEGRATION

The rapid development of World Wide Web has dramatically changed the way in which information is managed and accessed. The information in Web is increasing at a striking speed. At present, there are more than 7,500 terabytes (or 4 billion web pages) of information in Web.

- *J.Sharmila, Bharathidasan University College(W), orathanadu, Taminadu,India.*
- *Dr.A.Subramani, KSR College Of Engineering,Tamilnadu,India.*

Web information has covered all domains of human activities. This opened the opportunity for users to benefit from the available data. So Web is being concerned more and more.

Web can be divided into Surface Web and Deep Web. Traditional search engines create their indices by crawling Surface Web pages. Surface Web is the Web pages that are static and linked to other pages, while Deep Web refers to the Web pages created dynamically as the result of a specific search. Traditional search engines can not "see" or retrieve content in the Deep Web. On average, Deep Web receives fifty per cent greater monthly traffic than Surface Web. According to the survey released by UIUC in 2004, there are more than 300,000 Web databases and 450,000 query interfaces available at that time, and the two figures are still increasing quickly. Besides the scale of Web databases, the contents in Web databases are spanning well across all topics. Deep Web is being the largest growing category of new information on the Internet. The study on Deep Web will be one of the hottest areas in research.

2.1 Deep Web Integration

More and more accessible databases are available in the Web. In order to provide people a unified access to these Web databases and achieve information from them automatically, a comprehensive solution for Web database integration is proposed. The following figure is the architecture of the solution. This solution includes three primary modules: integrated interface generation module, query processing module and results processing module

Integrated interface generation module: Produce an integrated interface over the query interfaces of the Web databases to be integrated. There are four components in this module. The functions of them are described as follows:

- Web database discovery: Search Web sites which have Web databases behind, and identify the query interfaces among the Web pages in these Web sites.

- Query interface schema extraction: Extract the attributes in query interfaces, and the meta-information about each attribute.

- Web database clustering by topic: Cluster all discovered Web databases into different groups. The Web databases in each group belong to the same topic.

- Interface integration: Given the Web databases in the same topic, merge the same semantic attributes in different query interfaces into a global attribute, and finally form an integrated interface.

2.2 Query processing module:

Process a user's query filled in integrated interface, and submit the query to each Web databases. There are three components in this module. The functions of them are described as follows:
Web database selection: Select appropriate Web databases for a user's query in order to get the satisfying results at minimal cost.

2.3 Query translation:

Try to translate the query on integrated interface equivalently into a set of local queries on the query interfaces of Web databases.
Query submission: Analyze the submission approaches of local query interfaces, and submit each local query automatically.

2.4 Result processing module:

Extract the query results achieved from Web databases, and merge the results together under a global schema. There are three components in this module. The functions of them are described as follows:
Result extraction: Identify and extract the pure results from the response pages returned by Web databases.
Result Annotation: Append the proper semantics for the extracted results.

2.5 Result merging:

Merge the results extracted from different Web databases together under a global schema.

3. WEB INTEGRATION CHALLENGES

3.1 Time Consuming

Despite constrained budgets and limited resources, most user requirements today require their technology groups to deliver solutions faster than ever before. In the world of integration, determining which API to use for integration is a critical step. Industry approaches such as web services promote a uniform programmatic interface to all applications, yet all of the normal development issues remain. Web services or even hand-coded solutions have to be built and thoroughly tested before they are deployed. Normally this involves changing the application to expose its logic, a resource intensive and time consuming development effort. More pragmatically, this approach is only possible when your company has ownership of the entire application.

3.2 Impractical

For web applications outside of the enterprise, the modifications required by traditional integration methods will often not be possible. Even when a partner or customer is willing to open up their systems, the ability to integrate may be impractical. As a result, most business-critical projects requiring integration to external applications are extremely difficult at best, impossible at worst for most organizations, this can translate directly to missed business opportunities and increased operational costs.

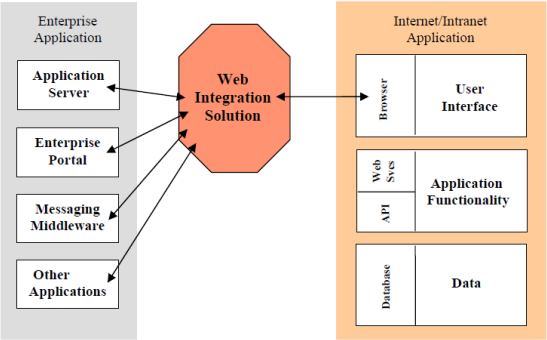
3.3 Expensive

Traditional application integration relies upon having total control of the application, so it can be modified internally. This often results in significant costs, since most application integration projects represent very large development efforts using highly skilled staff. Inherent in these larger projects is a high risk of delay and a possible impact on overall infrastructure of the organization.

4. WEB INTEGRATION OPTIONS

Below is a general illustration of potential integration points into a target web application. In concept, connectivity can take place at one of three different layers of an application:

- User Interface
- Application Functionality
- Data Access layer



The Application Functionality layer is the traditional place for programmatic access to an application, and is done by integrating with an API or using web services. It provides direct access to the business logic of an application. In some instances, however, an API does not exist and integration must be done by modifying the application code itself.

Integration at the Data Access layer is typically done through connections directly to the underlying databases. This is the most efficient approach to obtain data available through the application, yet because it by-passes and circumvents the business logic of the application, it is used only for data inquiry and aggregation requirements. True application integration must take place using the logic of the target. By utilizing the applications transaction logic, for functions such as updates, the integrity of the new data being added or modified can be insured. Since the User Interface layer, or presentation layer, was designed initially to allow an operator to interact with the application, it provides a standard and universal API providing access to both the application business logic and to all application data.

5. WEB INTEGRATION SOLUTIONS

Web Integration is a unique and innovative approach to the challenges of integrating and service enabling existing browser-accessible applications. Web Integration can be done quickly, iteratively, at low cost and with relatively modest skill requirements. In addition, because it uses the presentation layer as its API, it is by nature non-intrusive. The key factor with Web Integration is that most production applications already have an HTML based interface that provides access to both functionality and data. Most newly developed web applications even expose logic that has been written into the presentation layer itself.

Although this web interface was intended for end users, Web Integration can turn the browser into a well-defined programmatic interface that exposes the full functionality and data of the application. As the figure above shows, a Web Integration Server accesses the presentation layer of an application through the browser interface. This server can then make one or more of the following access points available: a web service, a modified browser interface, or an API into the application's logic and data.

Using Web Integration, other applications would have the capability to access the full functionality and data of the application, as if the application had been developed to provide an open interface. By accessing the application through the user interface, the power and functionality of these applications can be re-utilized in sophisticated integration solutions, such as:

- Enterprise Portals - content and functionality of existing web-enabled applications can be used in an enterprise portal.
- Composite Applications- functionality from any existing application can be combined to create a new application that automates business processes
- Web Services- any web-enabled application can be turned into a web service
- Data Collection- data available via any web-enabled application can be aggregated and moved to a new system, such as a content management system.
- Market Intelligence- information from competitors, media providers, government databases, etc., can be collected on a regular, scheduled basis for market intelligence purposes.
- Automation- in general, Web Integration is exceptional for programmatically automating normal operator tasks requiring the flow of information between web-enabled applications.

6. ONTOLOGIES FOR DATA INTEGRATION

Ontologies have been extensively used in data integration systems because they provide an explicit and machine-understandable conceptualization of a domain. They have been used in one of the three following ways,

6.1 Single ontology approach:

All source schemas are directly related to a shared global ontology that provides a uniform interface to the user. However, this approach requires that all sources have nearly the same view on a domain, with the same level of granularity. A typical example of a system using this approach is SIMS.

6.2 Multiple ontology approach

Each data source is described by its own (local) ontology separately. Instead of using a common ontology, local ontologies are mapped to each other. For this purpose, an additional representation formalism is necessary for defining the inter-ontology mappings. The OBSERVER system is an example of this approach.

6.3 Hybrid ontology approach.

A combination of the two preceding approaches is used. First, a local ontology is built for each source schema, which, however, is not mapped to other local ontologies, but to a global shared ontology. New sources can be easily added with no need for modifying existing mappings. Our layered framework is an example of this approach.

We identify the following five uses of ontologies in data integration:

1) Metadata Representation. Metadata (i.e., source schemas) in each data source can be explicitly represented by a local ontology, using a single language.

2) Global Conceptualization.The global ontology provides a conceptual view over the schematically-heterogeneous source schemas.

3) Support for High-level Queries. Given a high-level view of the sources, as provided by a global ontology, the user can formulate a query without specific knowledge of the different data sources. The query is then rewritten into queries over the sources, based on the semantic mappings between the global and local ontologies.

4) Declarative Mediation. Query processing in a hybrid peer-to-peer system uses the global ontology as a declarative mediator for query rewriting between peers.

5) Mapping Support. A thesaurus, formalized in terms of an ontology, can be used for the mapping process to facilitate its automation.

7. WEB INTEGRATION BENEFITS

Web Integration has a number of important benefits compared to the other more traditional integration approaches:

1) Lower cost

Using Web Integration, the effort required becomes much less expensive compared to traditional integration. There are numerous reasons, including lower skills for developers, no change to existing applications, and no infrastructure changes to network.

2) Non-intrusive

Web Integration is done non-intrusively, thereby lowering the risk and impact of the entire integration project. Because there are no architectural changes required, it is often easier to justify cross-enterprise projects. This benefit extends even further to those external applications where the user interface is the only available option for integration.

3) Faster development

Since the browser interface is well understood by both the end user and the programmer, application design becomes much easier and less prone to error.

4) Faster overall integrations

Even complex Web Integration projects can be completed in weeks rather than months. Companies can gain competitive advantages by leveraging their existing enterprise applications more quickly than their competitors.

5) Lower skill requirements

A traditional integration project requires highly skilled development staff. In depth knowledge of the applications and application integration techniques is required. With Web Integration technologies, development personnel with basic programming experience and web application knowledge can do a superior job. The need for high-skilled expensive programmers can be obviated.

6) Potentially lower risk.

Web Integration allows for shorter and more cost-effective implementation cycles. The initial integration can often be up and running quickly and further integration can be accomplished once results from the initial integrations have been proven. This allows companies to try out new business opportunities at lower risk than using traditional methods.

8. CONCLUSION

The Integration of Web Data Sources is a large research field. While it shares many problems with the traditional data integration, it also has its own set of problems because of the characteristics of the web data sources. Web data integration research is evolving with the development of the Web as well as of the Data Integration methods. Currently web data integration only solves the problems relating to answering queries, but not with transactions between sources. The reason may be that most nowadays sources are autonomous. However, building a full featured data integration system over the Web could also a good way since it can take the advantages of Web infrastructure and matured technologies.

9. REFERENCES

[1] B. Amann, C. Beeri, I. Fundulaki, and M. Scholl. Ontology-Based Integration of XML Web Resources. In

Proceedings of the 1st International Semantic Web Conference (ISWC 2002), pages 117–131, 2002.

[2] M. Arenas, V. Kantere, A. Kementsietsidis, I. Kiringa, R. J. Miller, and J. Mylopoulos. The Hyperion Project: From Data Integration to Data Coordination. SIGMOD Record, 32(3):53–38, 2003.

[3] S. Bergamaschi, F. Guerra, and M. Vincini. A Peer-to-Peer Information System for the Semantic Web.In Proceedings of the International Workshop on Agents and Peer-to-Peer Computing (AP2PC 2003), July 2003.

[4] P. A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zaihrayeu. Data Management for Peer-to-Peer Computing: A Vision. In WebDB 2002, pages 89–94, 2002.

[5] I. F. Cruz, H. Xiao, and F. Hsu.Peer-to-Peer Semantic Integration of XML and RDF Data Sources.In The 3rd International Workshop on Agents and Peer-to-Peer Computing (AP2PC 2004), July 2004.

[6] S. Decker, S. Melnik, F. van Harmelen, D. Fensel, M. C. A. Klein, J. Broekstra, M. Erdmann, and I. Horrocks. The Semantic Web: The Roles of XML and RDF. IEEE Internet Computing, 4(5):63–74, 2000.

[7] T. R. Gruber. A Translation Approach to Portable Ontology Specifications.Knowledge Acquisition, 5(2):199–220, 1993.

WEBSITES

http://www.w3.org/TR/rdf-primer/
http://www.w3.org/TR/ws-arch/#resource_oriented_model
http://www.w3.org/TR/soap/
http://www.w3.org/2001/sw/
http://en.wikipedia.org/wiki/Web_syndication
http://developer.yahoo.com/search/rest.html