Mediators handle new data types emerging from the use of structuring methods like XML. Find out what mediators are and how they work.

David A. Grossman, Steven M. Beitzel, Eric C. Jensen, and Ophir Frieder

IIT Intranet Mediator: Bringing Data Together on a Corporate Intranet

Historically, the worlds of unstructured and structured data have required separate types of searches, generally following two common approaches. Searches for information in unstructured data sources, such as text documents, have relied on the first approach, information retrieval. Users enter search terms as a data query, and application programs using specialized application programming interfaces search unstructured data sources for the occurrence of these terms. Such a search can return text—the data itself—or, as in a Web search, return only the data's location or site. Users must then read the text or go to each site and locate the search terms to determine whether the results are in fact relevant to the actual query.

For structured-source searches, the second common approach, the application programs search highly structured data within one specific source to return a specific answer to a user's query. This data is usually privately owned and accessed. A user searching an airline reservation system or requesting a quote from an online travel portal is performing a structured search against a structured information source.

This type of source, commonly referred to as a database management system or data warehouse, is usually constructed and searched using relational database systems, such as Oracle and IBM DB2. These systems typically search one data source at a time. A desired set of facts often resides in individual databases across multiple sources, however, so, private businesses that own such data often integrate their individual data sources. This integration facilitates queries whose answers require more than one factual component, but it is expensive. In addition, the integrated source can remain underutilized without proper query formulation or extensive data source knowledge.

A hybrid data type, known as semistructured data, bridges the worlds of unstructured and structured data. Existing semistructured data formats such as XML (Extensible Markup Language) attempt to unite the virtues of structured and unstructured data by imposing some body of structure on a collection of free text. These formats accomplish structuring without the rigidity usually present in a relational database. As a result, semistructured data affords users a readable data format that also contains extractable structured information.

As collections of semistructured data increase in popularity, it is clear that existing search technologies must adapt to support them. Mediators are one promising technique now under development.

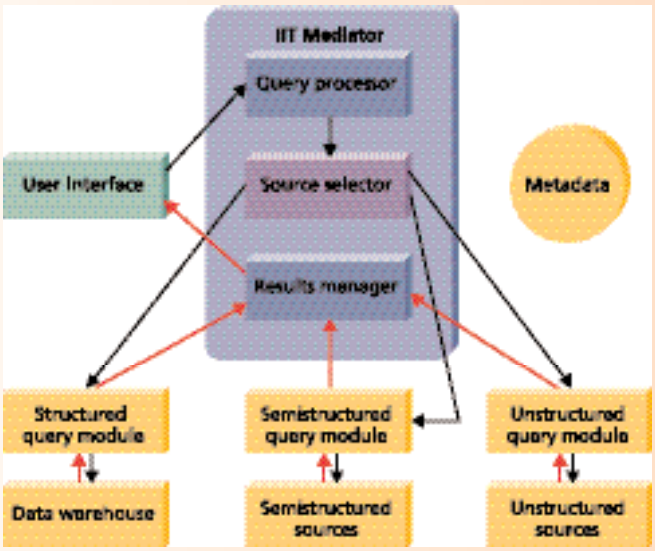
WHAT ARE MEDIATORS?

A *mediator* is a software module that interacts with a user and a variety of data sources to provide one-stop shopping for an organization's data (B. Ludäscher, A. Gupta, M.E. Martone, "Knowledge-

Inside

1520-9202/02/$10.00 © 2002 IEEE

Figure 1. IIT Mediator architecture.



Based Mediation and XML-Based Information Integration," *17th Int'l. Conf. Data Engineering*, IEEE CS Press, Los Alamitos, Calif., 2001). This approach is especially attractive when taken over a large, heterogeneous system of data, because the cost of source integration for such a system is high. Intranet and digital libraries could benefit from mediators because they handle myriad data types and must treat all data appropriately, according to its type.

For example, submitted to a search engine instead of a structured search, the user query "What are the three best sushi restaurants in Chicago?" likely results in a search for the word "three." A mediator, on the other hand, recognizes that this query needs to access structured information, then examines its available sources for a structured repository that contains the desired information. The mediator also submits requests to unstructured and semistructured sources. The result of this unified search effort is twofold: the mediator supplies a suggested answer to the user's question based on available structured information, and it also returns a set of related links to other potentially useful, related information. The mediator can retrieve this information from semistructured and unstructured sources, such as XML documents and plain text.

For our restaurant query, we might expect a mediator to return a ranked list of the top three Chicago sushi restaurants (perhaps retrieved from a restaurant ratings database) and links to food critics' reviews for various Chicago sushi restaurants.

A mediator's key goal is to simply answer natural-language questions, deriving answers from sources that contain structured, semistructured, and unstructured data in a particular domain. Various efforts, like those conducted as part of the National Institute of Standards and Technology's annual Text Retrieval Conference, focus on unstructured sources. The IIT Intranet Mediator extends these efforts to include structured and semistructured sources as well.

DATA SOURCES

Mediators aim to answer questions from either a virtual or physical data warehouse. Virtual data warehouses are perhaps best suited to Internet-centered uses. Physical data warehouse are easier to implement in intranet environments, in which a single company or small group controls the information.

Virtual data warehouse

A virtual data warehouse (VDW) does not exist physically. All data are stored across the network, as in the Internet, hosted by a variety of different databases. A user sends a query to the VDW via a mediator, which in turn accesses a single, unified schema. The schema indicates how to obtain each datum in the entire VDW. For example, a book-focused mediator might query multiple sources located across the Internet with "Who wrote *The Art of Computer Programming*?" A schema integrator then attempts to reconcile representational differences between the various sites with relevant information at query execution time.

Such dynamic reconciliation is appealing because data remains local to its site of origin and need not be copied. The mediator searches the data in a distributed manner, which means that a single query can search data from multiple sites. Control of the data remains under local administration, so privacy issues are easier to handle. VDW efforts are of primary research interest, and numerous efforts focus on them.

Physical data warehouse

A physical data warehouse (PDW) requires the replication of structured data from several sources (Lou Agosta, *The Essential Guide to Data Warehousing*, Prentice Hall, Upper Saddle River, N.J., 2000). Unlike a VDW, PDWs exist physically, each acting as a master source that contains all the data gathered from several sources. Database administrators under the direction of a warehouse project manager build PDWs using an extract, transform, and load (ETL) process, which migrates data from the disparate source databases to a central data warehouse. This process generates summary information from the data and also stores it in the warehouse. PDWs perform schema reconciliation at load time. This entire process occurs before the posing of any queries to the system. This approach devotes greater care and processing time to schema reconciliation (it is an offline process), enhancing accuracy. Further, user

response time is shorter than with a VDW because no schema reconciliation occurs at query time. However, these advantages come at the expense of data replication and loss of privacy. Thus, PDWs are generally feasible in intranet environments where data access falls under a single domain—that is, when queries come from, say, a single division within a company.

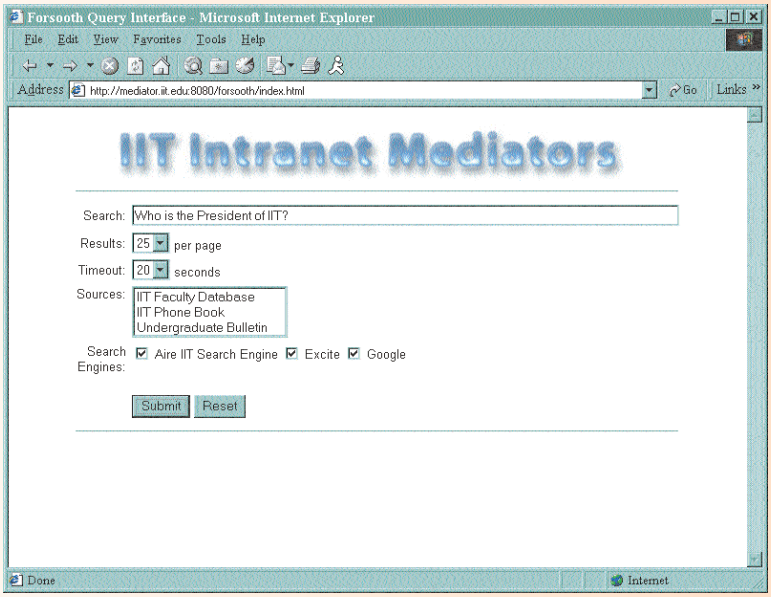
IIT MEDIATOR ARCHITECTURE

The IIT Mediator is an intranet mediator in use at the Illinois Institute of Technology. Figure 1 shows our mediator's high-level architecture.

User interface: Initial query

The user interface is both the starting and the ending point in the query process. It is responsible for accepting a query from the user, passing it on to the rest of the mediator, and displaying any results. Our user interface is a forms-based Web interface, similar to those driving many major Web search engines, with two notable improvements. First, we encourage users to enter their queries in complete, natural-language sentences (I. Androutsopoulos, G.D. Ritchie, and P. Thanisch, "Natural Language Interfaces to Databases—An Introduction," *Natural Language Eng.*, vol. 1, part 1, 1995, pp. 29-81). The query is then passed to the query processor for parsing and token identification. Second, we let users manually select which of our available sources to use during the query process. With these improvements, the user can issue very powerful queries on data searched by our mediator. Figure 2 shows a typical query screen.

Figure 2. IIT Intranet Mediator user interface, query input screen.



Metadata

To aid in source selection, mediators usually keep a store of metadata about each of their data sources (George A. Mihaila, Louiqa Raschid, and Maria Esther Vidal, "Query Evaluation for Source Selection and Ranking," *Third Int'l Workshop on the Web and Databases*; http://www.research.att.com/conf/webdb2000/PAPERS/6c.ps). Metadata, "data about data," provide the mediator with descriptive information about the sources that it oversees. The IIT Mediator keeps separate stores of metadata for each type of source that it has access to. For structured sources, Mediator tracks the top values for each attribute and also stores common synonyms for these values and attributes. For unstructured sources, it stores the top terms in each source along with their synonyms. Additionally, because semistructured data formats such as XML are hierarchical, a value must be associated with each path. The mediator takes this hierarchical element into account by storing the paths for each top value. This gives it a method for retrieving contextual information about a piece of semistructured data, which greatly aids the process of accurate query generation. When taken in aggregate, the information kept in the metadata aids both source selection and source-specific query generation.

Query processor

Using these sentences, natural-language analysis tools extract meaningful concepts from the query (I. Androutsopoulos, G.D. Ritchie, and P. Thanisch, "Natural Language Interfaces to Databases—An Introduction," *Natural Language Eng.*, vol. 1, part 1, 1995, pp. 29-81). The query processor first tags the query for parts of speech **and then calls** a natural-language parser, which identifies the query's key linguistic components.

Source selector

When the query processor passes a query to the Mediator source selection component, a probabilistic parts-of-speech tagger analyzes the query. This analysis extracts meaningful query concepts in the hope of capturing some element of the user's information need. Once the tagger identifies these concepts, Mediator uses information from its store of metadata to determine whether any of the available sources contain values that match the discovered query concepts. A hierarchy of domain-specific rules exists in the source selector, giving it the ability to identify which source is relevant to a given query. We are currently developing a rule lan-

Resources

- *Extensible Markup Language (XML) 1.0 (Second Edition),* **W3C recommendation, 6 Oct. 2000; http://www.w3.org/TR/2000/REC-xml-20001006: This document is the basis for XML.**
- *XML-QL: A Query Language for XML,* **Alin Deutsch and colleagues, NOTE-xml-ql-19980819, World Wide Web Consortium, 19 Aug. 1998; http://www.w3.org/TR/1998/NOTE-xml-ql-19980819/: XML-QL includes a rich query language, and our mediator can generate the necessary XML-QL requested by the natural-language query.**
- **"Database Techniques for the World Wide Web: A Survey," Daniela Florescu, Alon Levy, and Alberto Mendelzon, *SIGMOD Record*, vol. 27, no. 3, Mar. 1998, pp. 59-74: This paper summarizes several techniques for using databases on the Web and contains a useful discussion on the problems involved in on-line schema integration.**
- **Annual Text Retrieval Conference (TREC), National Institute of Standards and Technology, http://trec.nist.gov: TREC continues to be a key conference in the text retrieval field.**

guage and testing the ability of these rules to scale to complex domains. The mediator then uses this information to refine the list of matching sources and determine which sources are most appropriate for the query posed. It adds any sources that the user has manually selected to the list of appropriate sources and then dispatches the query to the query modules corresponding to each appropriate structured and semistructured source.

Currently, Mediator sends the query in its natural-language form to all available unstructured sources, essentially performing a metasearch. The idea behind this approach is that the Mediator results, in addition to possibly containing an answer to the user's query, will at least be no worse than those obtained from a traditional metasearch (Daniel Dreilinger and Adele E. Howe, "Experiences with Selecting Search Engines Using Metasearch," *ACM Trans. Information Systems*, vol. 15, no. 3, July 1997, pp. 195-222).

Query modules

For each available source in Mediator, there is a query module. At present, there are structured, semistructured, and unstructured query modules. The query module components take the high-level notion of a query concept and translate it into an appropriate query for their particular source. These modules run in parallel and in multiple execution threads. For unstructured sources, the query-posing process consists of passing the natural-language query to the interface for the unstructured source. This interface can be an application programming interface, such as a locally available information retrieval engine like our Advanced Information Retrieval Engine (Abdur Chowdhury and colleagues, "IIT TREC-9: Entity Based Feedback with Fusion," *Overview of the Ninth Text Retrieval Conf.*, NIST special publication 500-249, July 2001; http://trec.nist.gov/pubs/trec9/t9_proceedings.html) or a Web-based interface for a remote Web search engine, such as Google or Excite.

For structured sources, the query module is responsible for translating queries into the appropriate structured query language (A.N. De Roeck and colleagues, "A Formal Approach to Translating English into SQL," *Aspects of Databases, Proc. Ninth British Conf. Databases*, Butterworth-Heinmann, Woburn, Mass., 1991). Typically, most relational database management systems express the query in SQL (Standard Query Language). The query modules for semistructured sources, such as a collection of XML documents, must also translate queries. In this case, we translate the query from natural language into a popular XML query language, such as XML-QL. The query translation uses elements of Mediator's metadata,

which helps determine the values and conditions required for a properly formatted query. For all sources, after query modules pose the queries, they return results to the Mediator's results manager component.

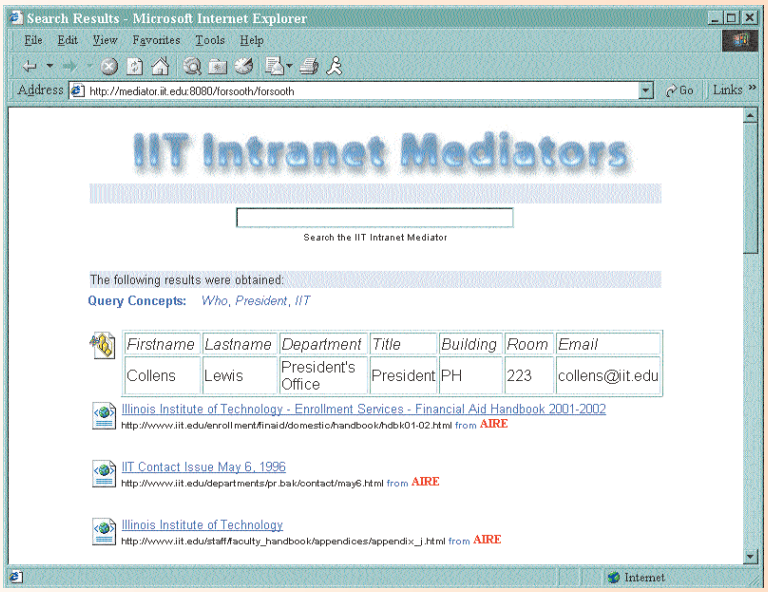
Results manager

Query modules are asynchronous—they do not all run at the same time and some can finish well before others. Their independent operation introduces interesting side effects. Results from the various sources do not necessarily arrive in a predictable order, and sometimes not in a reasonable time frame. Hence, we conducted a detailed analysis of various methods for managing the asynchronous arrival of results. Based on this analysis, we chose an approach that facilitates an improved ranking scheme in the results manager and ultimately more timely access to results sets.

Numerous other mediators take different approaches (Isabel F. Cruz and Kimberly M. James, "User Interface for Distributed Multimedia Database Querying with Mediator Supported Refinement," 26 Jan. 1999, http://www.cs.wpi.edu/~beez/Papers/IDEAS99.ps). For example, Inquirus asynchronously displays results (Eric J. Glover and colleagues, "Architecture of a Metasearch Engine that Supports User Information Needs," *Proc. Eighth Int'l Conf. Information Knowledge Management* (CIKM 99), ACM Press, New York, 1999). This strategy is effective in the sense that users need not wait for results, but it has the potential for suboptimal relevance ranking. In Inquirus, highly relevant documents might arrive after it has already done a ranking, introducing the problem of how to merge late-arriving, relevant documents into the existing ranked set. To alleviate this problem, an improved prototype, Inquirus 2, has two display windows, with one window constantly reranking documents. This avoids the problem of late-arriving, highly relevant documents, but now presents a confusing view to users. This system reranks documents in front of their eyes, making it difficult to comprehend the ranked list of results.

To achieve an optimal time/ranking tradeoff, our system waits for a short, fixed time period of t seconds for results to arrive. After t seconds, Mediator ranks the results it has and returns them to the user interface for display. Simultaneously, it continues retrieving documents in asynchronous threads of execution. As documents arrive, Mediator stores them in a cache. When the user hits the Next button, the results manager removes duplicates and shows the documents that arrived while the user was viewing the first set.

Figure 3. IIT Intranet Mediator user interface, query results screen.



User interface: Results display

Once some results become available, Mediator is faced with the issue of how to distinguish the display of structured results from that of semistructured and unstructured results. Our resolution of this issue is based on a set of assumptions. First, we assume that answers returned by structured sources are likely to be more relevant than mere documents or sites from by unstructured and semistructured sources. This situation is typically the case because data stored in structured repositories are generally pruned, processed, and hopefully verified prior to insertion into the database.

In contrast, documents, both semistructured and unstructured, are generally bulk loaded into their sources with few or no data quality measures. Based on this assumption, we display results obtained from structured sources first and present them as "answers" to the user's query.

We acknowledge, however, that in spite of structured data verification, our structured answers can still be incorrect. Thus, we also display a related-information section, as Figure 3 shows. In this section, we display summaries of ranked results from semistructured and unstructured sources and provide links from these summaries to the actual documents. The advantage of this approach is that users can quickly see the set of candidate answers to a question like "Who is the President of IIT?" In addition,

they can also see related links to relevant unstructured and semistructured information. This information might include, for example, a link to the IIT president's home page or an XML document containing contact information for IIT senior officers.

The IIT Mediator still requires further research and development to support wide use. Several issues have yet to be resolved. These issues include the feasibility of a physical data warehouse—we want to understand what it would take to build one on, say, even an intranet scale.

As the volume of data in warehouses grows, the global schema becomes more complex, and the source selector component has a more difficult task. In the future, we wish to explore several possibilities for augmenting the intelligence and capabilities of the source selector. Possible approaches include expanding our store of metadata, using information extraction techniques to improve concept identification in user queries, and using machine learning and data mining techniques in addition to rule sets for each data source.

Additional work would help automate the addition of new data sources to Mediator. For a mediator to become maximally useful, it must be easy to add new data. To this end, we are currently developing tools that allow digital-library administrators to quickly and easily add new data sources to a mediator. In their present form, these tools allow metadata information to be automatically generated for new sources, but someone must still hand-code the query module components for the source. Clearly, it is important to examine the possibility of automatically generating the code to query a source based on the source's characteristics.

Finally, the user interface is another area with great potential for future work. We are presently unaware of other user interfaces that tell the user "we think the answer is *x*" to answer questions and that also tell the user "for other related information, see *y*." This type of user interface seems reasonable for a digital library, but user studies would explore and validate such an interface's effectiveness. Finally, future work could improve the runtime efficiency of the source selector and results manager. ■

David Grossman *is an assistant professor of computer science at the Illinois Institute of Technology. Contact him at dagr@ir.iit.edu.*

Steve Beitzel *and* ***Eric Jensen*** *are PhD students in the Information Retrieval Laboratory at the Illinois Institute of Technology.*

Ophir Frieder *is the IITRI Professor of computer science at the Illinois Institute of Technology.*

We gratefully acknowledge work done by Michael Saelee and the support of the National Science Foundation.