Building Agents for Internet-based Supply Chain Integration

Craig A. Knoblock
University of Southern California
Information Sciences Institute and
Integrated Media Systems Center
4676 Admiralty Way
Marina del Rey, CA 90292-6695
Knoblock@isi.edu

Steven Minton
University of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695
Minton@isi.edu

ABSTRACT

As supply chains become more and more dynamic, the problem of rapidly integrating the data sources of new suppliers is becoming increasingly important. This paper describes the Ariadne system, which can be used to quickly provide access to data sources of new supplier and requires only that the data of the suppliers is available on the Web. Our approach does not require reengineering the individual systems to work together, which is both costly and time consuming.

Keywords

Agents, information integration, supply chain integration.

1. Introduction

The Internet has effectively solved one of the major hurdles in information integration: the connectivity problem. It is now very easy to obtain access to information resources via the Web, including databases and knowledge bases. This has spawned many new opportunities both within organizations (via intranets) and between organizations (via extranets). However, integrating multiple heterogeneous information sources is still extremely difficult, because sources often use different data formats, terminology, and ontological distinctions. Overcoming the connectivity problem has brought these deeper issues into the foreground.

We have developed a very general system called Ariadne (Knoblock et al., 1998) for rapidly building agents that integrate information from multiple heterogeneous sources, including databases, web sites, and programs. One of the difficult aspects of supply chain integration is that members of the supply chain must all agree on data interchange standards, and each must install software capable of communicating with their partners. Our system can simplify this process considerably. Members of the supply chain can independently specify what information to import from their partners, and how it should be represented. Partners need not agree on a single standard data format or representation, nor install externally produced software behind their firewalls. Our technology enables each organization, if it desires, to maintain its own representations and its own distinct view of the supply chain. Standards can be incrementally adopted as the supply chain evolves.

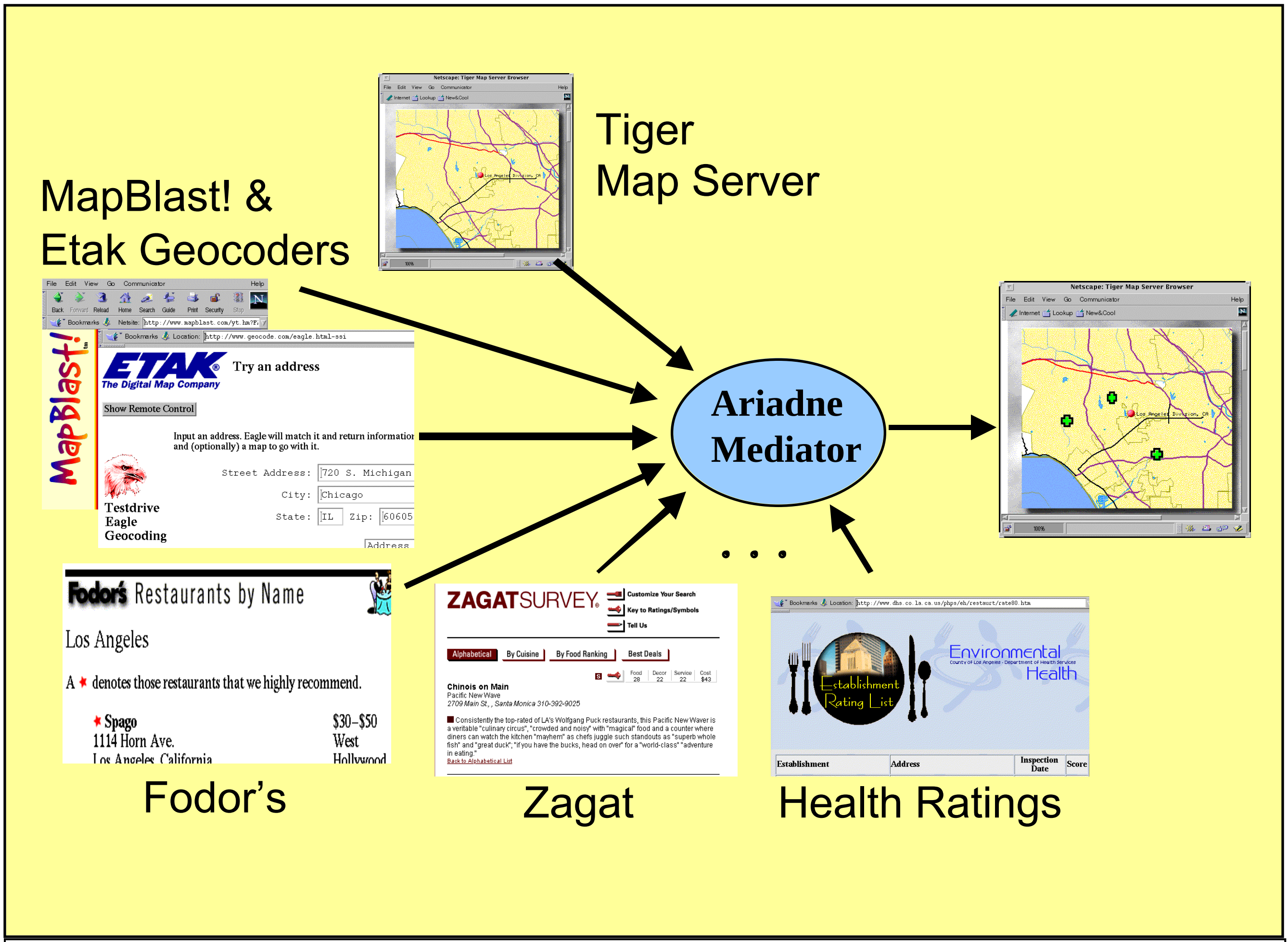
Because Ariadne employs automated induction techniques for integrating new sites, it is extremely easy to set up and maintain relatively sophisticated integrated applications. Thus, organizations in the supply chain obtain the best of all possible worlds: they gain the benefits of supply chain participation without incurring significant software re-engineering costs, and without being dependent on other organizations to adopt their standards.

Ariadne is most relevant to supply chain monitoring for large-scale networks of suppliers, where products are being continually, and rapidly, re-engineered or reconfigured. Consider personal computers, for example. PC manufacturers assemble their products from numerous components produced by a variety of suppliers. PC's are highly configurable; typically any given PC manufacturer offers a wide variety of models, each with different components often from different suppliers. The market changes rapidly due to technical innovation. In six months, a manufacturer might modify their product line significantly. For any given manufacturer, the supply chain may be quite fluid, in the sense that different suppliers may be added or removed from the chain in a relatively short amount of time.

This type of situation presents difficulties for the traditional centralized approach, since the cost of tightly integrating suppliers into the chain, and continually modifying the chain, is relatively high (for all parties). In contrast, our technology is well-suited to a fluid supply chain, since the distributed information system can be modified relatively easily. Furthermore, and perhaps more importantly, our technology allows for a variety of innovative applications. Imagine a supply chain with a completely integrated documentation system. Such a system would enable an engineer, help desk operator, customer or distributor to type in a product ID, and access the latest documentation for any component or subcomponent of that product, customized for their needs (an engineer or help desk operator would have different needs than a customer). Or imagine a product configuration system that would enable a customer to "design" his or her own product by selecting components and subcomponents. In addition to automatically validating configurations, the system could enable customers to explore how long it would take to manufacture the product based on each supplier's latest schedule. These applications are within the state of the art, but they are difficult and expensive to implement precisely because of the cost of integrating supplier's information systems. They are particularly problematic for fluid supply chains, where suppliers come and go. If we can develop systems that enable suppliers to be integrated into off-the-shelf systems at relatively low cost, then such applications will become commonplace.

2. Background

Ariadne is based on the SIMS (Single Interface to Multiple Sources) Information Mediator (Arens, Knoblock & Shen, 1996; Arens, Chee, Hsu & Knoblock, 1993) SIMS enables users to obtain information from multiple heterogeneous information sources. The framework consists of two parts: 1) a query planner/executor that determines how to efficiently process a query given the set of available information sources



Try an address

Input an address. Eagle will match it and return information and (optionally) a map to go with it.

Street Address: 720 S. Michigan
City: Chicago
State: IL Zip: 60605

Fodor's Restaurants by Name

Los Angeles

A ★ denotes those restaurants that we highly recommend.

★ Spago $30–$50
1114 Horn Ave. West
Los Angeles, California Hollywood

Food	Decor	Service	Cost
28	22	22	$43

Chinois on Main
Pacific New Wave
2709 Main St., , Santa Monica 310-392-9025

Consistently the top-rated of LA's Wolfgang Puck restaurants, this Pacific New Waver is a veritable "culinary circus", "crowded and noisy" with "magical" food and a counter where diners can watch the kitchen "mayhem" as chefs juggle such standouts as "superb whole fish" and "great duck"; "if you have the bucks, head on over" for a "world-class" "adventure in eating."

Figure 1: Architecture of the Restaurant Application

and 2) wrappers that provide uniform access to the information sources so that they can be queried as if they were all SQL databases.

An important idea underlying SIMS is that for each application there is a unifying *domain model* that provides a single ontology for the application. The domain model is used to describe the contents of each information source. Given a query in terms of the domain model, the system dynamically selects an appropriate set of sources and then generates a plan to efficiently produce the requested data.

Information mediators were originally developed for integrating information in databases (Wiederhold, 1996). Applying the mediator framework to the Web environment solves the difficult problem of gaining access to real-world data sources. The Web provides the underlying communication layer that makes it easy to set up a mediator system, because it is typically much easier to get access to Web data sources than to the underlying databases systems. In addition, the Web environment means that users who want to build their own mediator application need no expertise in installing, maintaining, and accessing databases.

Ariadne is a Web-based version of the SIMS mediator architecture. The Ariadne project's goal has been to make it simple for users to create their own specialized information agents. We have developed technology for rapidly constructing agents to extract, query, and integrate data from web sources, databases, and programs. The system includes tools for constructing wrappers that make it possible access these disparate types of sources, and the mediator technology required to dynamically and efficiently answer queries using these sources.

A simple example, shown in Figure 1 illustrates how Ariadne can be used to provide integrated access to multiple information sources. Numerous web sites provide reviews on restaurants, such as Zagats, Fodors, and CuisineNet, but none are comprehensive, and checking each site can be time consuming. In addition, information from other sources can be useful in selecting a restaurant. For example, the LA County Health Department publishes the health rating of all restaurants in the county, and the U.S. Census bureau provides a map server that will show selected locations. Using Ariadne, we can integrate these web sources relatively easily to create an

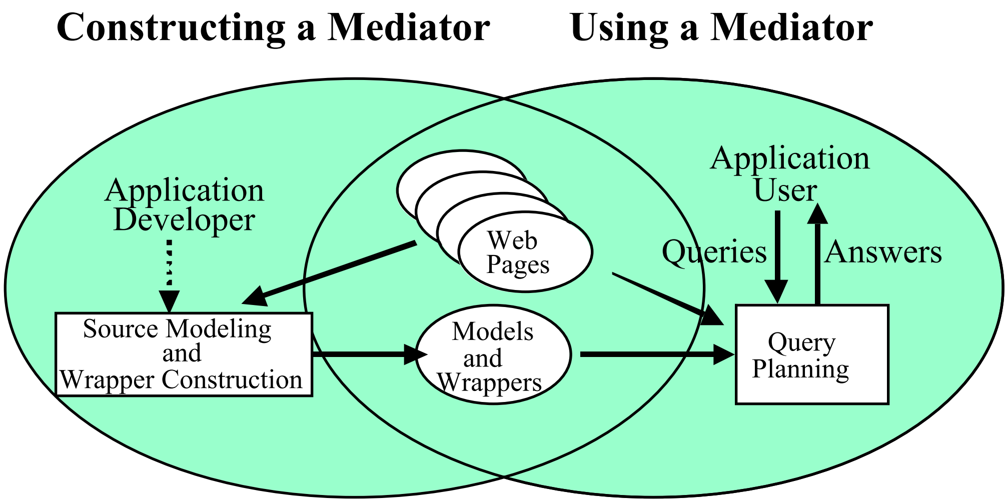


Figure 2: General Framework for Building an Application

application where people could search these sources to create a map showing the restaurants that meet their requirements.

With such an application, a user can pose requests that would, for instance, generate a map listing all the seafood restaurants in Santa Monica that have an “A” health rating and whose typical meal costs less than $30. The resulting map lets the user click on the individual restaurants to see the restaurant critic reviews. (In practice, we do not support natural language, so queries are either expressed in a structured query language or are entered through a graphical user interface.) The integration process that Ariadne facilitates can be complex. For example, to actually place a restaurant on a map requires the restaurant’s latitude and longitude, which is not usually listed in a review site, but can be determined by running an online geocoder, such as Etak, which takes a street address and returns the latitude and longitude.

Figure 2 outlines our general framework. We assume that a user building an application has identified a set of information sources he or she wants to integrate. These might be both publicly available sources, such as web sites, as well as a user’s personal sources, such as his or her own databases. For each source, the developer uses Ariadne to generate a wrapper for extracting information from that source. The source is then linked into an application specific, unified domain model. Once the mediator is constructed, users can query the mediator as if the sources were all in a single database. Ariadne will efficiently retrieve the requested information, hiding the planning and retrieval process details from the user.

Figure 3 shows a simplified fragment of a domain model that includes the restaurant application. The domain model actually covers a more general entertainment application that we have developed which includes movie theaters as well as restaurants. The domain model is represented using the Loom knowledge representation language (MacGregor, 1988). A model consists of a set of classes (e.g., restaurant, theater, address) and relations between these classes (e.g., ISA, Address-of). The model also includes a set of information sources (shown as databases in the figure) that are linked to the various classes. To preserve clarity the figure shows only a few of the classes and information sources for this application.

2.1 Research Issues

Our recent work (Knoblock, Minton et al., 1998) has focused primarily on extending our approach to database integration so that we can integrate information from multiple organizations via the Internet. Ariadne includes all of SIMS capabilities for accessing and integrating multiple databases, and in addition, includes technology specifically designed for the web environment. Below we focus on four new capabilities of the Ariadne system that illustrate how our research has extended beyond existing database integration techniques to address problems that are unique to the Web.

2.1.1 Converting Semistructured Data into Structured data

One of the challenges associated with integrating web sources into an application is that the format of web sources is not explicitly specified. Often sources consist of HTML (or plain ASCII) pages that are *semi-structured*, i.e., the format can be partially captured by a concise grammar, though some natural language may be included. (This definition excludes completely unstructured natural language texts). In order to integrate such sources into an Ariadne application, we must be able to query the sources as if they were databases. This is done using a wrapper, which is a piece of software that interprets a request (expressed in SQL or some other structured language) against a Web source and returns a structured reply (a set of tuples). These wrappers let the mediator both locate the Web pages that contain the desired information and extract the specific data off a page. The huge number of evolving Web sources makes manual construction of wrappers expensive, so Ariadne includes tools for rapidly building and maintaining wrappers for web sources (in addition to tools for creating wrappers for databases and programs).

Consider our restaurant mediator example. To extract data from the Zagats restaurant review site, a user would need to build two wrappers. The first lets the system extract the information from an index page, which lists all of the restaurants and contains the URLs to the restaurant review pages. The second wrapper extracts the detailed data about the restaurant from the page containing the address, phone number, review, rating, and price. With these wrappers, the mediator can answer queries to Zagats, such as “find the price and review of Spago” or “give me the list of all restaurants that are reviewed in Zagats”.

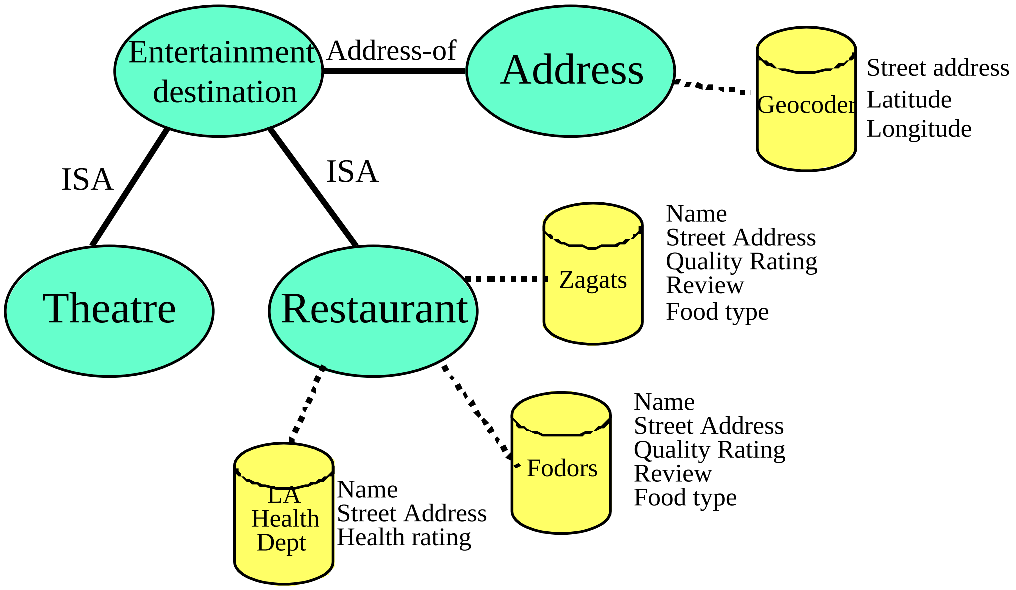


Figure 3: Fragment of Domain Model for the Restaurant Application

The core of our wrapper technology consists of the Stalker inductive-learning system (Muslea, Minton & Knoblock, 1999), which learns a set of extraction rules for pulling information off a page. The user trains the system by using a GUI where he or she can mark up example pages to show the system what information it should extract from each page. Stalker can learn rules from a relatively small number of examples by exploiting the fact that there are typically "landmarks" on a page that help users visually locate information. To build the wrapper for Zagats' restaurant review pages, the user would train the inductive learning system by selecting some example restaurant review pages, and on each example page, highlighting the restaurant's address, phone number, rating, review, etc. The inductive learning system can then learn extraction rules for each field. The learning system has been empirically shown to produce high-accuracy rules using only a few training examples per field.

2.1.2 Planning Techniques for Integrating Data

One of the fundamental advantages of the SIMS/Ariadne technology is that highly expressive SQL queries can be answered, even though an application may involve integrating a large number of underlying sources. We have found that web applications can be particularly challenging because the number of sources to be integrated is relatively large. (Before the Internet became popular, it was difficult to gain access to more than a few databases at a time.) Also, unlike relational databases, web sources often have restrictions on how a source can be accessed, such as a geocoder that takes the street address returns the geographic coordinates, but cannot take the geographic coordinates and return the street address.

Ariadne breaks down query processing into a preprocessing phase and a query-planning phase. In the first phase, the system determines the possible ways of combining the available sources to answer a query. Because sources might be overlapping—an attribute may be available from several sources—or replicated, the system must determine an appropriate combination of sources that can answer the query. The Ariadne source-selection algorithm (Ambite, et al., 1998) preprocesses the domain model so that the system can efficiently and dynamically select sources based on the classes and attributes mentioned in the query.

In the second phase, Ariadne generates a plan using a method called Planning-by-Rewriting (Ambite & Knoblock, 1997, Ambite & Knoblock, 1998) . This approach takes an initial, suboptimal plan and attempts to improve it by applying rewriting rules. With query planning, producing an initial, suboptimal plan is straightforward – the difficult part is finding an efficient plan. The rewriting process iteratively improves the initial query plan using a local search process that can change both the sources used to answer a query and the order of the operations on the data.

In our restaurant selection example, to answer queries that cover all restaurants, the system would need to integrate data from multiple sources (wrappers) for each restaurant review site and filter the resulting restaurant data based on the search parameters. The mediator would then geocode the addresses to place the data on a map. The plans for performing these operations might involve many steps, with many possible orderings and opportunities to exploit parallelism to minimize the overall time to obtain the data. Our planning approach provides a tractable approach to producing large, high-quality information-integration plans.

2.1.3 Providing Fast Access to Slow Web Sources

Accessing and extracting data from distributed Internet sources is generally much slower than retrieving information from local databases. Because the amount of data might be huge and the remote sources are frequently being updated, simply warehousing all of the data is not usually a practical option. Instead, we have developed an approach to selectively materializing (storing locally) critical pieces of data that let the mediator efficiently perform the integration task. The materialized data might be portions of the data from an individual source or the result of integrating data from multiple sources.

To decide what information to store locally, the algorithm carries out a utility analysis, taking several factors into account. First, we consider the queries that have been run against a mediator application. This lets the system focus on the portions of the data that will have the greatest impact on the most queries. Next, we consider both the frequency of updates to the sources and the application's requirements for getting the most recent information (as specified by a user). For example, in the restaurant application, even though reviews might change occasionally, providing information that is current within a week is probably satisfactory. But, in a finance application, providing the latest stock price would likely be critical. Finally, we consider the sources' organization and structure. For example, the system can only get the latitude and longitude from the geocoder by providing the street address. If the application lets a user request the restaurants located within a region of a map, it could be very expensive to figure out which restaurants are in that region because the system would need to geocode each restaurant to determine whether it falls within the region. Materializing the restaurant addresses and their corresponding geocodes avoids a costly lookup.

Once the system decides to materialize a set of information, the materialized data simply becomes another information source for the mediator. This is an elegant solution because it meshes well with our mediator framework, where the planner dynamically selects the sources and the plans that can most efficiently produce the requested data. In the restaurant example, if the system decides to materialize address and geocode, it can use the locally stored data to determine which restaurants could possibly fall within a region for a map-based query.

2.1.4 Resolving Inconsistencies Across Sources

Within a single site, entities—such as people, places, countries, or companies—are usually named consistently. However, across sites, the same entities might be referred to with different names. For example, one restaurant review site might refer to a restaurant as Art's Deli and another site might call it Art's Delicatessen. Or, one site might use California Pizza Kitchen and another site could use the abbreviation CPK. To make sense of data that spans multiple sites, our system must be able to recognize and resolve these inconsistencies.

In our approach, we select a primary source for an entity's name and then provide a mapping from that source to each of the other sources that use a different naming scheme. The Ariadne architecture lets us represent the mapping itself as simply another wrapped information source. Specifically, we can create a mapping table, which specifies for each entry in one data source what the equivalent entity is called in another data source. Alternatively, if the mapping is computable, Ariadne can represent the mapping by a mapping function, which is a program that converts one form into another form. Once the mapping (table or function) is represented as an information source, the planner will automatically use the source to translate from one terminology to another whenever it is required by the query.

We are currently developing a semi-automated method for building mapping tables and functions by analyzing the underlying data in advance. The basic idea is to use information-retrieval techniques (as in Cohen, 1998) to provide an initial mapping and then use additional data in the sources to resolve any remaining ambiguities via statistical learning methods. For example, restaurants are best matched up by considering name, street address, and phone number, but not by using a field such as city because a restaurant in Hollywood could be listed as either being in Hollywood or Los Angeles and different sites list them differently

In addition to handling naming inconsistencies, mapping tables and functions can be useful when two sources rely on different, but related, ontologies. For example, in Los Angeles, one restaurant source might list the city that a restaurant is located in while another might list the neighborhood (e.g., Manhattan Beach vs. South Bay). A mapping table enables us to convert from cites to neighborhoods. Similarly, in a supply chain application, one source might list the number of days until a shipment is due, while another source might list the number of weeks. A mapping function enables us to convert from one frame of reference to the other.

3. Related Work

There is large body of relevant literature by other researchers on information integration (Wiederhold, 1996), but the most closely related work focuses specifically on the problems of information integration on the Web, such as Information Manifold (Levy et al., 1996), Occam (Kwok & Weld, 1996), Infomaster (Genesereth et al. 1997), and InfoSleuth (Bayardo97). These systems focus on a variety of issues, including the problems of representing and selecting a relevant set of sources to answer a query, handling binding patterns, and resolving discrepancies among sources. All of this work is directly relevant to Ariadne, but these systems have not focused on rapid integration of new sources

Another closely related body of work is on the extraction of data from Web sources (Hammer et al., 1997; Doorenbos et al. 1997; Kushmerick, 1997). The focus of all of these systems is on building wrappers for semi-structured sources. The systems either take a template-based specification of a source, as in (Hammer et al., 1997) or learn the structure of the source by example and then compile a wrapper that provides access to the source, as in (Kushmerick, 1997). Our work on inducing wrappers takes the latter approach. The induction method is not only very general, but is also integrated into the larger Ariadne development system so that the learned wrappers can be used directly by the query planner.

4. Discussion

The infrastructure of the Web eliminates the problems of gaining access to distributed sources of data. This now makes it possible to build very fluid supply chains where suppliers publish the required data on the Web, which can then be directly integrated into the supply chain software of other companies. In this paper, we have described how the Ariadne information mediator makes it possible to rapidly assemble these dynamic supply chains. The current system has been applied in a variety of application domains and we continue to research ways to make it faster and easier to integrate new sources of data. The emerging XML standard will accelerate this process by eliminating the need for specialized wrappers in some application areas, although it will not completely solve the problem since not all sources will be in XML and the integration problems will still remain.

5. References

J.L. Ambite and C.A.Knoblock (1997) Planning by rewriting: Efficiently generating high-quality plans. In Proceedings of the National Conference on Artificial Intelligence

J.L. Ambite and C.A. Knoblock. (1998) Flexible and scalable query planning in distributed and heterogeneous environments. Proceedings of the Fourth International Conference on Artificial Intelligence Planning Systems, 1998.

J.L. Ambite, C.A. Knoblock, I. Muslea, and A. Philpot (1998) Compiling source descriptions for efficient and flexible information integration, Technical report, Information Sciences Institute, University of Southern California, 1998.

Y. Arens, C.A. Knoblock and C. Hsu (1996) Query Processing in the SIMS Information Mediator. In Advanced Planning Technology, edited by A. Tate. The AAAI Press, Menlo Park, CA.

Y Arens, CA. Knoblock, and W. Shen. (1996) Query Reformulation for Dynamic Information Integration. Journal of Intelligent Information Systems, Special Issue on Intelligent Information Integration. Vol. 6, No. 2/3

Bayardo Jr., R.J.; Bohrer, W.; Brice, R.; Cichocki, A.; Fowler, J.; Helal, A.; Kashyap, V.; Ksiezyk, T.; Martin, G.; Nodine, M.; Rashid, M.; Rusinkiewicz, M.; Shea, R.; Unnikrishnan, C.; Unruh, A.; and Woelk, D. (1997). Infosleuth: Agent-based semantic integration of information in open and dynamic envi- ronments. In Proceedings of ACM SIGMOD-97.

W.W. Cohen (1998) Integration of Heterogeneous Databases without Common Domains Using Queries Based on Textual Similarity, Proc. ACM Sigmod-98, ACM Press

R.B. Doorenbos, O. Etzioni, and D.S. Weld (1997) “A Scalable Comparison-Shopping Agent for the World-Wide Web,” Proc. First Int’l Conf. Autonomous Agents, AAAI Press

M.S Fox and M. Gruninger (1998), "Enterprise Modelling", AI Magazine, AAAI Press, Fall 1998, pp. 109-121.

M.S. Fox, and Gruninger, M., (1997), "On Ontologies and Enterprise Modelling", International Conference on Enterprise Integration Modelling Technology 97

M.R. Genesereth, A.M. Keller, and O.M. Duschka, (1997) “Infomaster: An Information Integration System, Proc. ACM Sigmod Int’l Conf. Management of Data, ACM Press

J. Hammer, H. Garcia-Molina, S. Nestorov, R. Yerneni, M. Breunig, and V.Vassalos (1997) Template-based wrappers in the TSIMMIS system. In Proceedings of ACM SIGMOD-97.

N. Kushmerick, N. (1997) Wrapper Induction for Infor- mation Extraction. PhD thesis, Computer Science Dept., University of Washington.

C.T. Kwok and D.S. Weld (1996) Planning to gather information. In Proceedings of AAAI-96.

C.A. Knoblock, S. Minton, J.L. Ambite, N. Ashish, P.J. Modi, I. Muslea, A.G. Philpot and S.Tejada, (1998) Modeling Web Sources for Information Integration, Proceedings of the National Conference on Artificial Intelligence

A. Y. Levy, A. Rajaraman and J.J. Ordille (1996) Query-answering algorithms for information agents. In Proceedings of the National Conference on Artificial Intelligence.

A.Y. Levy, A. Rajaraman, and J.J. Ordille, (1996) “Querying Heterogeneous Information Sources Using Source Descriptions,” Proc. 22nd Very Large Databases Conf., Morgan Kaufmann, San Francisco

R.M. MacGregor (1988) A Deductive Pattern Matcher, in Proceedings of the National Conference on Artificial Intelligence (AAAI 88), Morgan Kaufmann, San Mateo, CA,

I, Muslea, S. Minton and C.A. Knoblock (1999) A Hierarchical Approach to Wrapper Induction, Proceedings of the 3rd International Conference on Autonomous Agents

C. Schlenoff, R. Ivester, and A. Knutilla, (1998) A Robust Process Ontology for Manufacturing Systems Integration, ,Proceedings of the 2nd International Conference on Engineering Design and Automation

A. Tate (1998) "Roots of SPAR - Shared Planning and Activity Representation", The Knowledge Engineering Review, Vol. 13(1), pp. 121-128, Special Issue on "Putting Ontologies to Use" (eds. Uschold, M. and Tate, A.), Cambridge University Press.

G. Wiederhold (1996) Intelligent Integration of Information. Kluwer Academic Publishers.