

Multimodal Integration of Disparate Information Sources with Attribution

(Position Paper)

Thomas Y. Lee and Stéphane Bressan
MIT-Sloan School of Management
E53-320, 50 Memorial Drive
Cambridge, MA, 02139, USA
{tlee,sbressan}@mit.edu

Abstract: In this paper, we describe results from a preliminary effort to better understand issues of attribution that arise from the application of knowledge-based systems to the blending of structured querying and document retrieval over disparate (heterogeneous, distributed) information sources. The term multimodal integration is introduced to refer to such a blending of structured querying and document retrieval.

1. Introduction

While the World Wide Web offers a rich pool of resources, the mixture of media types, and diverse storage and access methods presents users with the challenge of divining useful information from the ether. One promising avenue for structuring the tangled Web calls for merging knowledge-based systems with traditional document retrieval in a multimodal paradigm using the Mediation reference architecture introduced in (Wiederhold, 92). In addition to the Context Interchange research project, from which this work stems, several projects (e.g. Arens & Knobloch, 92; Garcia-Molina, 95; Tomasic et al., 95; Levy et al., 95; Papakonstantinou et al., 96; Duschka & Genesereth, 97).

The Context Interchange (COIN) Project (Goh et al., 94), in particular, is developing a model (Goh et al. 97; Bressan et al., 97c), a prototype (Bressan et al. 97a; Bressan et al., 97b), and tools (Bressan and Lee, 97; Bressan and Bonnet 97), for the knowledge-based semantic integration of disparate (distributed and heterogeneous) information sources ranging from on-line databases to semi-structured Web services. From the perspective of a given data source, end-user, or intermediate application, context knowledge constitutes a declarative specification for how data is interpreted. By representing and reasoning about contexts, COIN's automated resolution of semantic conflicts enables a transparent access to heterogeneous information sources.

Two key issues that arise from the COIN approach for applying knowledge-based systems to blend structured querying and document retrieval are those of the extraction of structured information from documents, and attribution. Given the integration of disparate information sources, attribution asks how an end-user may discover which documents or databases support a given answer or value. To warrant attributing a specific source, however, assumes that the COIN system can even access said sources and extract meaningful data.

In this paper, we describe results from a preliminary effort to better understand issues surrounding attribution and extraction of data from documents that arise from the blending of the structured query and document retrieval (multimodal) paradigms in systems like COIN. Briefly, we have integrated some elements of the (Madnick and Wang, 90) attribution model for a federated, relational database into Web wrapping (Qu, 96; Bressan and Bonnet, 97; Bressan and Lee, 97), a strategy for extracting information from semi-structured documents which stems from the COIN information integration research. The resulting features constitute first steps towards resolving problems of attribution for knowledge-based system support of structured database querying over the Web.

In the next section, we present an example query, which demonstrates the features of our prototype implementation. We revisit the example throughout the remainder of the paper to help illustrate central concepts. In (Section 3), we expand on the issue of attribution and summarize key elements of the (Wang and Madnick, 90)

attribution model. (Section 4) recalls the fundamental principles behind the Web wrappers. (Section 5) describes our architecture and implementation. We conclude by discussing implications of our work for future research.¹

2. Example

One particularly promising application of the evolving, open information infrastructure has been financial planning and financial analysis. Not only do companies themselves host Web pages, but the exchanges themselves as well as a number of financial services companies post prices, earnings, industry-sector indices, portfolio management recommendations, and more. Consider a case where you are gathering financial recommendations for making stock purchases on the New York Stock Exchange (NYSE). Given the continuing Internet explosion, you might be particularly interested in viewing the current prices for recommended "strong buys" in the NYSE market segment of "data processing, software". If formulated in pseudo-SQL, the prospective query might ask:

```
select ticker, company_name, recommendation, and price from 'the Web'
where NYSE_market_sector = "data processing, software" and recommendation = "buy".
```

Unfortunately, the Web, as we know it today, does not lend itself to such a rich, structured, query language. Moreover, the diversity of available information is paralleled only by the formats in which data is stored and the means by which that data is retrieved. Beneath the overarching World Wide Web lies a myriad of flat files, file systems, and commercial database products. Unless all of this information is consolidated in a single data source, the reader has no recourse other than to first locate one or more sites from which the desired information may be accumulated and then to navigate each site in a piece-meal fashion to extract the desired information.

Our prototype presents a relational interface to a variety of Web sites preliminarily selected. For users seeking financial recommendations on NYSE securities, our prototype presents a 'virtual' relation (call this relation nyse) that exports the following schema:

nyse: nyse.Ticker, nyse.Company_name, nyse.Industry_sector, nyse.ShareOut, nyse.Recommendation nyse.Last_price

"Company_name" is the he 'official' name for each company listed on the NYSE. "Ticker" is the symbol of the company's security traded on the NYSE. "Industry_sector" is the sector classification name as defined by the NYSE. "Shareout" is the number of outstanding shares. It is updated to the state of the security at the close of the market on the prior business day. "Recommendation" is a numerical recommendation calculated as an average of Wall Street Brokers polled by USA-Today at the close of the market on the prior business day. Recommendations take the form of an integer ranking ranging from 1 to 5 where 1 indicates a "strong buy", 3 is a "hold", and "5" is a strong sell. "Last" is the most recent price of the security. This figure is guaranteed to be at _least_ 15 minutes delayed from the current price.

In the context of the virtual relation, 'nyse,' our original, prospective query may now be read as:

```
select nyse.Ticker, nyse.Company_name, nyse.Recommendation, nyse.Price from nyse
where nyse.Market_sector = "data processing, software" and nyse.recommendation < 2
```

This example query results in the following data table:

Ticker	SSW	CSC	...
Company_name	Sterling Software, Inc.	Computer Sciences Corporation	...
Recommendation	1.7	1.8	...
Price	32 7/16	77 7/8	...
Table valid as of Thu Aug 14 17:52:19 1997 cols =4; rows = 13			

Table 1. Database query result set (excerpt)

¹ Because this work is still in its infancy, we have chosen to provide references to related work into the relevant sections rather than to structure and synthesize a catalogue raisonné.

In addition, the system has recorded, for each data cell in the table, the document from which the data value stems. The following table represents the query results attributed to a list of sources:

Ticker	SSW <ul style="list-style-type: none"> http://www.nyse.com/public/data/symguide/symlst2.csv http://www.ultra.zacks.com/cgi-bin/ShowFreeCompRepUSAToday?ticker=SSW http://qs.cnnfn.com/cgi-bin/stockquote?symbols=SSW 	CSC <ul style="list-style-type: none"> http://www.nyse.com/public/data/symguide/symlst2.csv http://www.ultra.zacks.com/cgi-bin/ShowFreeCompRepUSAToday?ticker=CSC http://qs.cnnfn.com/cgi-bin/stockquote?symbols=CSC 	...
Company_name	Sterling Software, Inc. <ul style="list-style-type: none"> http://www.nyse.com/public/data/symguide/symlst2.csv 	Computer Sciences Corporation <ul style="list-style-type: none"> http://www.nyse.com/public/data/symguide/symlst2.csv 	...
Recommendation	1.7 <ul style="list-style-type: none"> http://www.ultra.zacks.com/cgi-bin/ShowFreeCompRepUSAToday?ticker=SSW 	1.8 <ul style="list-style-type: none"> http://www.ultra.zacks.com/cgi-bin/ShowFreeCompRepUSAToday?ticker=CSC 	...
Price	32 7/16 <ul style="list-style-type: none"> http://qs.cnnfn.com/cgi-bin/stockquote?symbols=SSW 	77 7/8 <ul style="list-style-type: none"> http://qs.cnnfn.com/cgi-bin/stockquote?symbols=CSC 	...
Table valid as of Thu Aug 14 17:52:19 1997 cols =4; rows = 13			

Table 2. Query result set with attribution (excerpt)

Each cell associates a value with a list. The list contains a URL for each Web document that was accessed during the query processing and from which the corresponding cell value was extracted. For example, in Table 1, the Ticker symbol "CSC" could be read from:

- <http://www.nyse.com/public/data/symguide/symlst2.csv>, the complete NYSE listing;
- <http://www.ultra.zacks.com/cgi-bin/ShowFreeCompRepUSAToday?ticker=CSC>, a page reporting analysts recommendation for the CSC security;
- and <http://qs.cnnfn.com/cgi-bin/stockquote?symbols=CSC>, a page reporting the latest performance figures of the CSC security in the NYSE.

whereas the Recommendation was found only at:

- <http://www.ultra.zacks.com/cgi-bin/ShowFreeCompRepUSAToday?ticker=CSC>.

The system is accessible from desktop applications (Excel Spreadsheets, Web browsers, etc.) through a variety of interfaces tuned for modern protocols (ODBC, HTTP, etc.). In particular, we have experimented with several Web interfaces to our prototype. They take advantage of the most recent hypertext features (Dynamic HTML, JavaScript) supported by the standard browsers to propose elegant mechanisms for navigating between data cells and values, the corresponding lists of sources, and the documents themselves.

In Summary, the Web wrappers support a relational abstraction for structured queries over multiple Web sites, and the integration of the attribution model enables our prototype to return a hybrid result table that associates every element of every tuple in the result set with its corresponding sources. This builds a bi-directional bridge between the Web documents and the structured data one can extract from them.

3. Attribution

In an abstract sense, attribution produces a parallel data structure of sources that complements the transparency provided by mediated access to disparate sources. Such a feature has broad appeal. For example, attribution helps to ensure that intellectual property protections are honored. Given transparent access to multiple sources,

attribution suggests one means for measuring data quality: any result is only as good as the sources that provided the inputs. Finally, attribution is useful in archive management or data warehousing for monitoring updates or tracking revisions.

From the standpoint of the application domains, this feature can benefit the many applications, which needs to process the large amounts of data and information available on public and corporate World Wide Web networks. For instance, the financial services industry is already using the Web as a resource for business intelligence and market research. Likewise, the scientific community is using the Web as a transparent interface to heterogeneous, distributed data sources for coordinating national and international research. An example, at a particularly large scale, is the Human Genome project's extensive use of Web-accessible archives for documenting and sharing progress.

We begin by defining attribution as the process of associating data elements or groups of data elements in the result of a structured query over heterogeneous and distributed information systems to the sources which contributed to retrieving the data element(s). In other words, attribution can be described as explication or as an abstraction of query execution in terms of the primary sources. From (Section 2), the "buy" recommendation on TSK is attributed to an average of Wall Street Broker Buy-Sell recommendations as reported by Zack's Investment Research² available through USA Today³.

However, this initial definition of attribution is vague and incomplete. For example, what is a source? Is an author's name adequate? A bibliographic reference? And what is the nature of an answer that requires attribution? Is it a hit-list of document handles as in a conventional Web query? Is it a data table for which rows or columns derive from one or more sources? Given our current conceptualization, attribution is defined in terms of a query framework (data model and language) as well as the nature of the sources (relations and tables or documents and hyper-links).

(Wang and Madnick, 1990) present an attribution model called data-tagging for a federated relational database model called the Polygen model. From the Data-tag model, we may derive further insight into both the nature of a 'source' and what an answer is that requires attribution. In the data-tagging model a source is a reference to one of the primary databases participating in the federated database.

A data-tag is a pair of sets [S] and [I]. A data-tag is associated to each data cell in the result table of a query. [S] is the set of references to the databases from which the result is drawn. [I] is the set of "intermediate" sources, i.e. the references to the databases that contributed to the selection of the data. From (Section 2), we queried for all stocks in the "data processing, software" category that received better than a "buy" recommendation. In this case, the set [I] in the data tag of the price values contains references to the sources of the recommendations. Indeed, although the recommendation is a "source" for the latest price of a particular security, it contributed (from the query constraint recommendation < 2) to selecting the tickers and latest prices. The recommendation is an "intermediate".

The data-tags associated to a value in a query result, and therefore, the sets [S] and [I], are defined inductively from the data-tag compositions occurring for each operator in the relational algebra of the Polygen model. The base step of the induction hypothesizes that, for data in the source databases, [S] is initialized to the singleton containing a reference to the database, and [I] is initialized to the empty set. The definition ensures that equivalent algebraic expressions of the query define the same attribution.

The main Polygen algebra operators are project, Cartesian product, restrict (selection), union, difference, and coalesce. Project, Cartesian product, and union, simply combine the tables of results and the respective data-tags, merging the [S] and [I] sets when duplicate elimination occurs. Restrict, however, propagates the elements of the sets [S] in the data-tags of the attributes involved in the restriction condition into the set [I] in the data-tag each attribute of the tuple (e.g. the propagation of the source of the recommendation in our example above). Restrict is the base for the definition of Theta-join and natural join. The natural join [Date, 87] is usually defined as the

² <http://www.zacks.com>

³ <http://www.usatoday.com>

composition of an equi-join (Theta-join where the condition operator is equality) and a project. However, in order to acknowledge the inherent symmetry of the projection (either one of or the other attribute involved in the restriction could be chosen), Wang and Madnick use a definition of the natural join based on the coalesce operator in place of the projection. Conceptually, coalesce merges the two attribute values. In consequence, they can give a definition of the corresponding propagation of [S] and [I] that respects this symmetry.

In the two following sections, after having outline the principles of the Web wrappers, we describe how we integrate the data-tag model in order to provide a mechanism for attribution in the process of the querying semi-structured Web documents. For the sake of simplicity we only consider the sets [S] of the data-tags.

4. Wrappers

Wrappers, as defined in (Wiederhold, 92), provide the physical connectivity and the first level of logical connectivity at the back end of the mediation architecture. At the logical level, they are in charge of providing an abstraction of the heterogeneity of the paradigms and models of the information sources. In the COIN architecture they provide a relational query interface to the external sources. A representative set of contributions about wrappers can be found in (Suciu, 97).

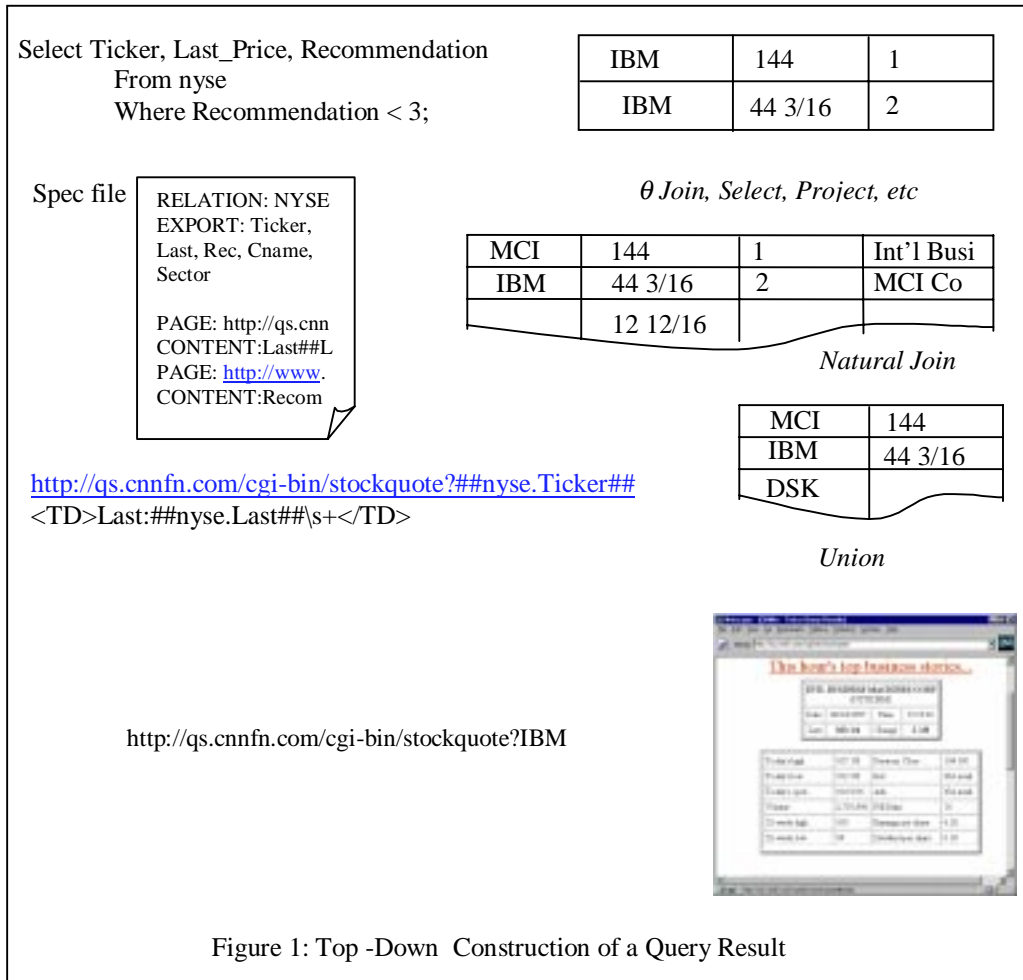
The Web wrappers, in particular, answer the challenge of enabling the easy and rapid integration of Web services as structured data sources in a mediation network. Their aim is not to be a universal tool capable of interfacing all kinds of Web sites including full-text or applets based services. Rather, they allow, by means of a simple specification language, to rapidly construct a relational interface to a reasonable number of semi-structured Web sites. By semi-structured, we refer to those numerous sites serving Web documents for which certain patterns identifying data elements can be recognized in the structure and the layout of the presentation. We assume a relative perennity of these patterns allowing the automatic identification and extraction of structured information from the document.

(Figure 1) illustrates the bottom up process of the construction of the answer to a query from Web documents. A program in the specification language, saved in a specification file, describes the patterns of the pages to interface, as well as the re-composition strategy, which defines a relational export schema. A page, in our terminology, is the association of a pattern for a HyperText Transfer Protocol (HTTP) message (the method, the Universal Resource Locator -URL-, and the object body) with patterns to be matched in the document source.

In the simplest case the pattern for the page message is the message itself (it does not contain variables). In such a case the page corresponds to a single document. However, some elements of the message may be variables. In such a case the page correspond to a set of documents: one for to each possible message generated by a substitution of the variables. Typically, this feature is used to describe documents dynamically generated via the common gateway interface (cgi) (generally, from a form based interface), or stored uniform collections of similar documents (for instance varying only in the file name). In our example a pattern for a page is “POST <http://www.ultra.zacks.com/cgi-bin/ShowFreeCompRepUSAToday?ticker=##Ticker##>”. Replacing the variable ##Ticker## by a ticker symbol defines an existing URL. The message corresponds to the request of an existing document: for instance, <http://www.ultra.zacks.com/cgi-bin/ShowFreeCompRepUSAToday?ticker=CSC>”. Patterns in the document are simple regular expressions, which are usually easy to construct from a rapid sight at a prototypical document source.

As in the example above, the variables in the patterns are associated to attributes of the relation being designed. Since patterns can receive several instantiations from both the message and the document, they virtually define a table of values, each column of which can be associated with an attribute of the relation. The co-occurrence of values in an instantiation of several variables in a page defines the rows of the table. The instance of the relation being designed by means of one or more pages is defined as the natural join of the individual tables corresponding

to each page respectively. The relation is designed under the Universal Relation concept: attributes of the same name are the same. The window function is the natural join⁴.



This constitutes a particularly simple and natural (non-exclusive) alternative to providing a rich but complex view definition language (e.g. Web wrappers of the TSIMMIS in (Suciu, 97)). Although it obviously limits the number of sites that can be wrapped, our practical experience has shown that it covers a reasonable number of situations of interest.

Given the declarative definition of the exported relations, the wrapper takes an input query in SQL over the exported schema, considers the corresponding specification files, and creates and executes a query execution plan. The query execution plan describes the various steps for accessing the documents, collecting data from the documents into individual tables, and combining the intermediary results into the answer to the query. The query plan is a tree of relational algebra operators (Theta-join, natural join select, project) and access operators (fetch and scan of a document). Under the semantics imposed by the choice of the window function (in our case the natural join), the wrapper attempts to minimize the number of documents accessed and the volume of intermediary information. Notice, however, that it may not be possible in the first place to generate a plan as no strategy may be found which guarantees that enough information is available to fully instantiate the message patterns and access the necessary documents. For instance the last prices and recommendations of the stocks in the industry sector "data processing, software" could not be accessed if the tickers for that sector were not listed in a document. This

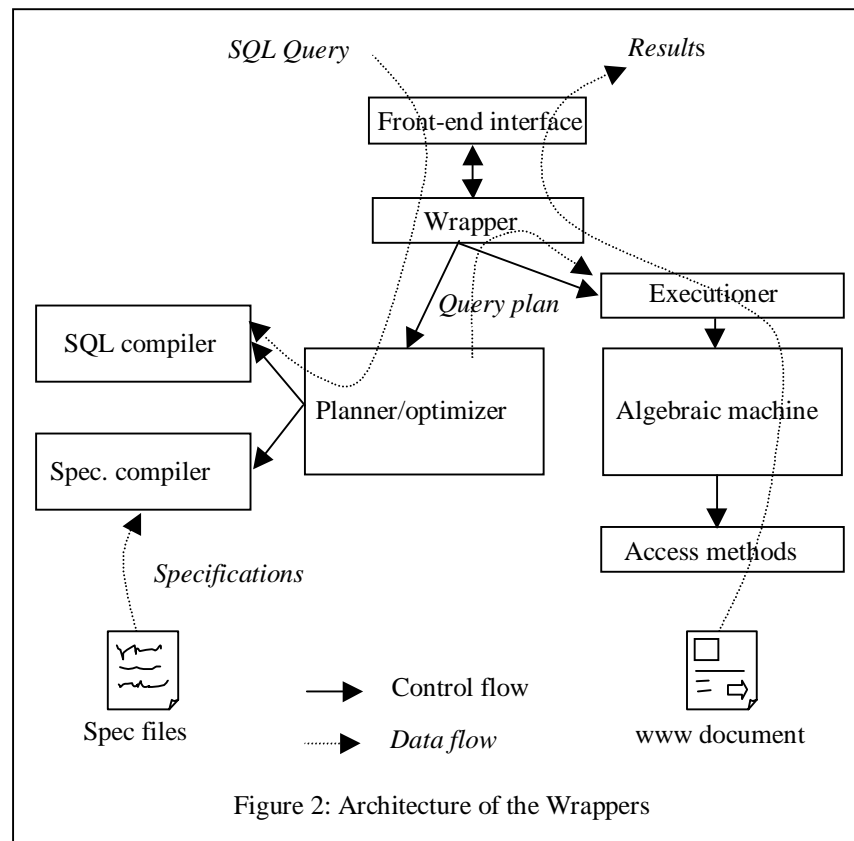
⁴ Alternative window functions could be considered in conjunction with integrity knowledge. They may lead to a higher potential for optimization and efficient evaluation.

part of the problem is called capability restriction. Efficient planning strategies under these circumstances have been studied in (Papakonstantinou et al. 95).

Thanks to the hypertext paradigm of the Web itself, the notion of service does not require that all documents reside on the same server. In fact in many applications, we have found convenient to consider as a single service a set of complementary services published by different providers. A relation can be defined over a set of documents originating from various sources. Although creating a very powerful integration paradigm, this retrospectively reinforces the needs for an attribution mechanism allowing to manage the diversity of sources from the results of a query back to the documents effectively involved in the construction of the result.

5. System architecture

We now describe the integration of the data-tag attribution model in a prototype implementation of the Web-wrappers. Because Web wrapping takes a document-centric approach, as illustrated in (Figure 1 and 2), our attribution model defines a source as a Web document. More precisely, as elaborated upon in (Section 4), a source is defined as the string that results from the evaluation of a fully instantiated HTTP message. This distinction is important because a URL may not correspond to a single, physical document. Instead, a URL, method, and message body may invoke a script to dynamically generate a response much as our prototype does. More formally, for a tuple T with value V_i corresponding to attribute A_i , the set $[S_i]$ is initialized to the fully instantiated HTTP message which returns the page from which V_i was extracted.



Illustrated in (Figure 2), the general wrapper architecture consists of a set of front-end interfaces, a set of compilation and planning modules, and a set of execution modules. The management of the data-tags only requires changes in the interfaces (for the presentation of the query results) and the execution modules (to implement the data-tags management associated with the relational operators).

The SQL compiler and Specification file compiler provide the planner/optimizer with a goal and an initial search space from which it formulates a query plan. The Planner implements standard strategies for the optimization of queries in multi-databases. However, it accounts for capability restriction. The leaves of the plan can be data from the query itself, or references to documents in the form of messages, which can either be instantiated or contain variables to be instantiated at run-time.

The relational operators in a query plan correspond to physical operators implemented in the algebraic machine. In general, a query execution is evaluated, in a set oriented manner, from the leaves to the root of the plan. Intermediary data are stored in a table data structure. The one exception to this bottom-up flow of data lies in the case of the join. The Algebraic machine implements the join in two ways. In the straightforward implementation, the join node combines two independently evaluated sub-plans and data flows bottom up. In some instances, however, we require the intermediary result from the left-hand-sub-plan of the join to inform the right-hand-sub-plan. In (Section 2), we might collect a list of Ticker symbols from one document and subsequently instantiate each Ticker symbol in an HTTP message to retrieve the latest price from another document. Called a join-scan and illustrated in (Fig. 3), this process is similar to the way in which semi-join filters operate in a distributed database.

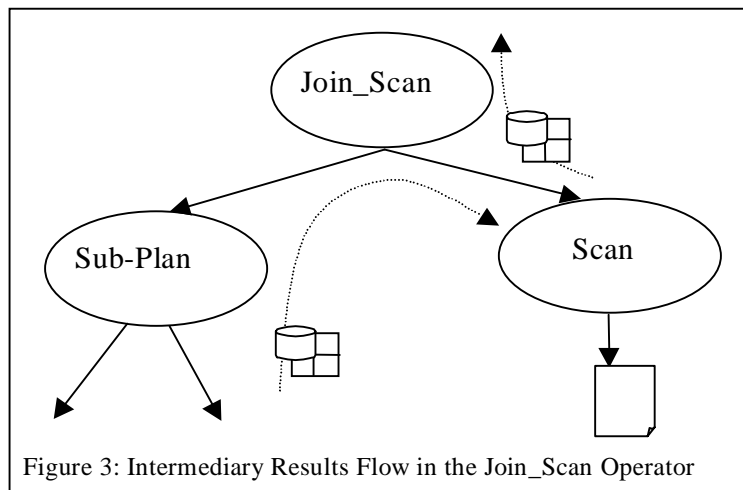


Figure 3: Intermediary Results Flow in the Join_Scan Operator

The algebraic machine required modifications not only to accommodate the new internal data structure to host the source sets [S], but also because each relational operator now must not only evaluate tuple values but also update their respective source sets. Specifically, recall that the wrapper implementation employs two different physical join operators that range over sets of tuples (tables). Because the two operators share a single, tuple-level physical operator, the join and join-scan operators exhibit the same semantics.

6. Conclusion

Having outlined our current implementation, we now turn to a number of questions and issues that were raised while developing and testing this prototype.

The first logical question might ask how to extend our current prototype to account for the management of the set of intermediary sources [I] introduced by (Wang and Madnick, 90). This does not present any conceptual difficulties for the current wrapper framework but does raise an implementation issue. Recall from (Section 3) that (Wang and Madnick, 90) rely on a conceptual difference between the theta join and the natural join in their Data-tag model for the definition of [I]. In the wrapper model, the Theta-join stems from an explicit join condition in the query, while the natural join is an inherent manifestation of the Universal Relation concept application (Section 4). Nevertheless, these two operators are implemented from the same physical operators: join and project (Section 5).

Therefore, in order to manage the set of intermediary sources we need to differentiate the specific physical operations by introducing the notion of coalesce.

A second question, which raised from the implementation, asks whether maintaining a parallel datastructure to manage the data-tags is necessary. It seems possible and appropriate to factorize the data-tags as often as possible (e.g. for rows, columns, or even arbitrary slices of the table of results). In fact, we anticipate that a relational model in non-first normal form (Abiteboul and Bidoit 84) would capture in a uniform manner the nesting, unnesting and necessary structuring operations to manipulate a richer attribution structure. This idea of factorization and distribution may also suggest to re-consider the notion of source reference itself. Recall (Table 2) from our example. Notice that a partially instantiated message with a well-placed wild-card or variable reference could minimize a large degree of duplication. In the recommendation column, the single HTTP message: `http://www.ultra.zacks.com/cgi-bin/ShowFreeCompRepUSAToday?ticker=##nyse.Ticker##` could be stored in lieu of the redundancy in repeating the string for every value in the column. In addition, as permitted by the structure of HTML documents, reference to multiple sources could be replaced by reference to new documents dynamically recomposed from fragments of the original documents.

Of course such a proposal leads to the conceptual issue of what it means to attribute a row, a column, or even a table for that matter. More formally, we seek to generalize our implementation into a mathematical model for attribution that seeks to map a query plan and the result to an attribution structure. Such a model should enable us to study properties of diverse attribution schemes.

Finally, we recall that our current conceptualization of attribution is tightly bound to a specific query framework and to specific sources. We would like to know whether more general notions of attribution exist. Must our concept of attribution for a query like the one posed in (Section 2) change when posed against a video library of interviews with brokers versus free-text newspaper articles versus structured databases? Is attribution a timed off-set into a video clip, a bibliographic reference to a newspaper article, a relation and attribute name? Does there exist solutions to deal with the issue of timeliness of the results as it appears in our example when a document is updated between the extraction of data and its consultation by the user? In asking this question, we revisit the issue of what constitutes a source and, by extension, ask what constitutes the attribution for a source.

8. References

- [Abiteboul and Bidoit 84] Abiteboul, S., and Bidoit, N. *Non First Normal Form Relations to Represent Hierarchical Organized Data*. PODS84. 1984.
- [Arens and Knobloch 92] Arens, Y. and Knobloch, C. *Planning and reformulating queries for semantically-modeled multidatabase*. Proc. of the Intl. Conf. on Information and Knowledge Management. 1992.
- [Bonnet and Bressan 97] Bonnet, Ph., and Bressan, S. *Extraction and Integration of Data from Semi-structured Documents into Business Applications* Submitted. 1997.
- [Bressan et al. 97a] Bressan, S., Fynn, K., Goh, C., Madnick, S., Pena, T., and Siegel, M. *Overview of a Prolog Implementation of the Context Interchange Mediator*. Proc. of the Intl. Conf. on Practical Applications of Prolog. 1997.
- [Bressan et al. 97b] Bressan, S., Fynn, K., Goh, C., Madnick, S., Pena, T., and Siegel, M. *The Context Interchange Mediator Prototype*. Proc of ACM-SIGMOD. (see also <http://context.mit.edu/demos/sigmod>). 1997.
- [Bressan et al. 97c] Bressan, S., Goh, C., Lee, T., Madnick, S., and Siegel, M. *A Procedure for the Mediation of Queries to Sources in Disparate Contexts*. To appear in Proc of the International Logic Programming Symposium. 1997.
- [Bressan and Lee 97] Bressan, S., and Lee, T. *Information Brokering on the World Wide Web*. To appear in the Proc. of the WebNet World Conf. 1997.
- [Date 87] Date, C. *An Introduction to Database Systems*. Addison Wesley. 1987.
- [Duschka and Genesereth 97] Duschka, O., and Genesereth, M. *Query Planning in Infomaster*. <http://infomaster.stanford.edu>. 1997.

- [**Garcia-Molina 95**] Garcia-Molina, H. *The TSIMMIS Approach to Mediation: Data Models and Languages*. Proc. of the Conf. on Next Generation Information Technologies and Systems. 1995.
- [**Goh et al. 94**] Goh, C., Madnick, S., Siegel, M. *Context Interchange: Overcoming the Challenges of Large-scale interoperable database systems in a dynamic environment*. Proc. of the Intl. Conf. On Information and Knowledge Management. 1994.
- [**Goh et al. 97**] Goh, C., Bressan, S., Madnick, S., and Siegel, M. *Context Interchange: New Features and Formalism for the Intelligent Integration of Information*. MIT-Sloan Working Paper #3941. 1997.
- [**Levy et al. 95**] Levy, A., Srivastava, D., and Kirk, T. *Data Model and Query Evaluation in Global Information Systems*. J. of Intelligent Information Systems. 1995.
- [**Wang and Madnick 90**] Wang, R. and Madnick, S. *A Polygen Model for Heterogeneous Database Systems: the Source Tagging Perspective*. Proc. Of the 16th Intl. Conf. On Very Large Databases. 1990.
- [**Papakonstantinou et al. 95**] Papakonstantinou, Y., Gupta, A., Garcia-Molina, H., and Ullman, J. *A Query Translation Scheme for Rapid Implementation of Wrappers*. Proc of the 4th Intl. Conf. on Deductive and Object-Oriented Databases. 1995.
- [**Papakonstantinou et al. 96**] Papakonstantinou, Y., Gupta, A., and Haas, L. *Capabilities-Based Query Rewriting in Mediator Systems*. Proc. of the 4th Intl. Conf. on Paralled and Distributed Information Systems. 1996.
- [**Qu 96**] Qu, J. *Web Wrapping in the Context Interchange project*. MIT Master thesis. Dpt. of Elect. Eng. 1996.
- [**Suciu 97**] Suciu, D. (Editor) *Proc. of the Workshop on Management of Semi-structured Data*. In Conj. with SIGMOD97. 1997.
- [**Tomasic et al. 95**] Tomasic, A., Rashid, L., and Valduriez, P. *Scaling Heterogeneous databases and the Design of DISCO*. Proc. of the Intl. Conf. on Distributed Computing Systems. 1995.
- [**Ullman 88**] Ullman, J. *Principles of Database and Knowledge-base Systems, Volume 1*. Computer Science Press, Rockville, MD. 1988.
- [**Wang and Madnick 88**] Wang, R., and madnick, S., *Connectivity among Information Systems*. Composite Information Systems Project 1. 1988.
- [**Wiederhold 92**] Wiederhold, G. *Mediation in the Architecture of Future Information Systems*. Computer, 23(3). 1992.