To appear in SIGMOD '97

The Context Interchange Mediator Prototype

S. Bressan, C.H. Goh, K. Fynn, M. Jakobisiak,
K. Hussein, H. Kon, T. Lee, S. Madnick, T. Pena,
J. Qu, A. Shum, M. Siegel

Sloan WP# 3940 CISL WP# 97-02
February 1997

The Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA 02142

The COntext INterchange Mediator Prototype*

S. Bressan, C. H. Goh†, K. Fynn, M. Jakobisiak, K. Hussein,
H. Kon, T. Lee, S. Madnick, T. Pena, J. Qu, A. Shum, M. Siegel
Sloan School of Management, Massachusetts Institute of Technology
Email: context@mit.edu

Abstract

The *Context Interchange* strategy presents a novel approach for mediated data access in which semantic conflicts among heterogeneous systems are not identified a priori, but are detected and reconciled by a *context mediator* through comparison of *contexts*. This paper reports on the implementation of a Context Interchange Prototype which provides a concrete demonstration of the features and benefits of this integration strategy.

1 The Context Interchange Strategy

The *Context Interchange* (COIN) project aims to develop tools and technologies for supporting access to information in heterogeneous and distributed systems. Our approach is founded on an integration strategy, called *Context Interchange* [GBMS96, SSR94]. The COIN strategy is novel because it provides for mediated data access *without* requiring semantic conflicts among heterogeneous systems to be identified a priori; instead, these disparities are detected and reconciled by a special-purpose *mediator* [Wie92], called a *context mediator*, through comparison of the *contexts* associated with sources and receivers engaged in data exchange.

Context mediation — the detection and reconciliation of conflicts which takes place during data exchange — is based on sound logical inferences. The representation and reasoning underlying the mediation strategy has been formally captured in a COIN *framework* [GBMS96], which is in turn build on a deductive and object-oriented data model of the family of *Frame-Logic* [KL89]. In addition to the standard abstraction features of any object-oriented formalism, the COIN data model provides built-in language features for capturing statements in a multi-theory framework: i.e., instead of requiring all statements to be consistent with one another, the logical statement are partitioned into collection of *context theories* such that all statements within a given context theory are consistent with one another, but any two statements drawn from two distinct context theories need not be so. The statements in a context theory provide an explicit codification of the implicit semantics of data in the corresponding "context".

For statements in a context theory to be meaningful in a different context, there needs to be a vocabulary common to all contexts, and a mapping that identify what individual data elements in a source refers to. The first takes the form of a *domain model*, which can be understood as a collection of "rich" types, or *semantic-types*. The latter is accomplished through a collection of *elevation axioms* which identify the elements of the source schema with the types in the domain model.

Queries in the COIN framework are source-specific: a user formulates a query identifying explicitly the sources and attributes referenced, *but under the assumption there are no conflicts between sources whatsoever*. The context mediator rewrites a query posed in a receiver's context into a *mediated* query where all potential conflicts are explicitly resolved. This rewriting, based on an *abductive* procedure [KK93], is accomplished by determining what conflicts exist and how they may be resolved by comparing relevant statements in the respective contexts.

The COIN strategy combines the best features of existing *loose-* and *tight-coupling* approaches [SL90] to semantic interoperability among autonomous and heterogeneous systems by allowing the complexity of the system to be harnessed in small chunks, by enabling sources and receivers to remain loosely-coupled to one another, and by sustaining an infrastructure for data integration. The integration approach is not only *non-intrusive* but also *scalable*, *extensible* and *accessible* [GMS94]. We claim that the approach is *scalable* because the complexity of creating and administering (maintaining) the interoperation services do not increase exponentially with the number of participating sources and receivers, since the addition of new sources or receivers requires only incremental instantiation of a new context (if one does not already exist). It is *extensible* because changes can be incorporated in a graceful manner in our framework: in particular, changes within any system can be effected by corresponding changes in local elevation axioms or context theory and do not have adverse effects on other parts of the larger system. Finally, the integration strategy being proposed here is *accessible* because it allows different kinds of queries to be supported while leveraging on the common knowledge structures in the system. A more detailed report on the kinds of queries and answers which can be handled is found in [GBMS96].

*This work is supported in part by DARPA and USAF/Rome Laboratory under contract F30602-93-C-0160.

†Financial support from the National University of Singapore is gratefully acknowledged.

2 The Prototype

Although our integration strategy may be applied to different application scenarios, we have chosen to develop the COIN prototype to illustrate the integration of databases and semi-structured information sources accessible from the Internet. This allows us to leverage on the large number of disparate information sources as well as the underlying network infrastructure for demonstrating the earlier claims (of scalability, extensibility, and accessibility) concerning the COIN strategy.

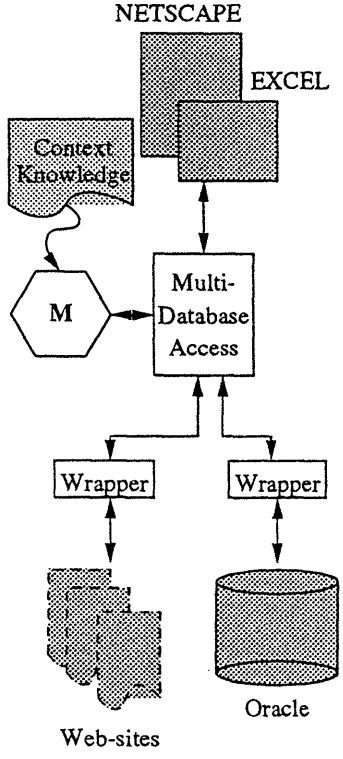

Figure 1: Architectural overview of the Context Interchange Prototype.

Figure 1 shows the architecture of the COIN Prototype. The sources we consider range from on-line databases (e.g. an Oracle database) to semi-structured Web-sites. Users and application programs (e.g. users of Web-browsers, spreadsheets, data-warehouses) have transparent access to the remote information sources (e.g. on-line databases, web-sites) through a server providing the mediation services.

Wrappers provide a uniform protocol for accessing corresponding sources and constitute the interface between the mediator processes and the sources. The wrappers are not merely communication gateways between the multi-database access engine and the sources, but they also provide a SQL interface to any source including the Web-sites and deliver answers to the queries in a relational table format. The Web wrapping technology we have developed [Qu96] is based on a high level declarative language for the specification of what information can be extracted. A program in this specification language defines a transition network corresponding to the possible transitions from one Web-page to another, and regular expressions corresponding to what information is located on a page.

On the receiver's side we have implemented an Application Programming Interface (API) of the family of the Object
DataBase Connectivity (ODBC) protocol. The protocol supporting this API is currently tunneled in the HyperText Transfer Protocol (HTTP) of the World Wide Web. The API can be used within any application with basic capabilities for Internet socket based communication. However, we have developed two types of ready-to-use interfaces: A HyperText Markup Language (HTML) Query-By-Example (QBE) and an ODBC driver which gives access to the mediation services to any Windows95 and WindowsNT ODBC compliant applications such as Microsoft Excel or Microsoft Access.

The multi-database access engine constitutes a front-end of dictionary and query services to the multiple wrapped sources. Its main functions are:

- Serving schema information such as names and attribute types of the table located in the various sources;
- Planning and optimizing the multi-source queries taking into account the sources capabilities as well as the execution and communication costs;
- Controlling the execution of the resulting query execution plan and executing the necessary local operations (e.g. joins across sources).

For the management of dictionary information and in order to handle large results or large sets of temporary data, the multi-database access engine uses two local secondary storages.

The mediation engine intercepts a query to the multi-database engine and rewrite it according to the context knowledge it has about the receiver and the sources involved. The rewritten query is usually a union of sub-queries corresponding respectively to the possible conflicts between the context assumptions and their resolution.

3 A Short Example of Mediation

Consider, for instance, the query "What are the names and revenues, of the companies whose revenue is bigger than their expenses?" Assume that such a query involves two sources and one relation in each source. The tables in the sources and the ancillary web source reporting currency exchange rates are shown on figure 2. The query is expressed in SQL:

```
SELECT  r1.cname, r1.revenue FROM r1, r2
WHERE r1.cname = r2.cname
AND r1.revenue > r2.expenses;
```

The above query, however, does not take into account the fact that data sources are administered independently and have different *contexts*: i.e., they may embody different assumptions on how information contained therein should be interpreted. For instance, the data reported in the two sources differ in the currencies and scale-factors of company financials (i.e., financial figures pertaining to the companies, which include revenue and expenses). Specifically, in Source 1, all company financials are reported using the currency shown and a scale-factor of 1; the only exception is when they are reported in Japanese Yen (JPY) in which case the scale-factor is 1000. Source 2 reports all company financials in USD using a scale-factor of 1. In the light of these remarks, the (empty) answer returned by executing Q1 is clearly not a "correct" answer since the revenue of

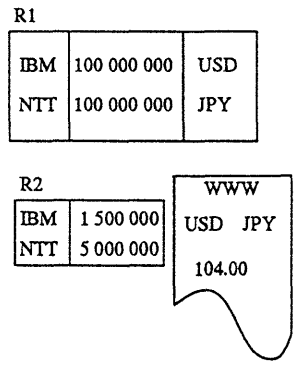

Figure 2: The relations R1 and R2, and the currency exchange Web source.

NTT (9,600,000 USD = 1,000,000 × 1,000 × 0.0096) is numerically larger than the expenses (5,000,000) reported in r2. The query is rewritten by the mediation engine into:

```
SELECT r1.cname, r1.revenue
FROM r1, r2
WHERE r1.currency = 'USD'
AND  r1.cname = r2.cname
AND r1.revenue > r2.expenses;
UNION
SELECT r1.cname, r1.revenue * 1000 * r3.rate
FROM  r1, r2, r3
WHERE r1.currency = 'JPY'
AND r1.cname = r2.cname
AND r3.fromCur = r1.currency
AND r3.toCur ='USD'
AND r1.revenue * 1000 * r3.rate > r2.expenses
UNION
SELECT r1.cname, r1.revenue * r3.rate
FROM r1, r2, r3
WHERE r1.currency <> 'USD'
AND  r1.currency <> 'JPY'
AND r3.fromCur = r1.currency
AND r3.toCur =  'USD'
AND r1.cname = r2.cname
AND r1.revenue * r3.rate > r2.expenses;
```

The mediated query considers all potential conflicts between relations r1 and r2 when comparing values of "revenue" and "expenses" as reported in the two different *contexts*. Moreover, the answers returned may be further transformed so that they conform to the *context* of the receiver. Thus in our example, the revenue of NTT will be reported as 9 600 000 as opposed to 1 000 000. More specifically, the three-part query shown above can be understood as follows. The first sub-query takes care of tuples for which revenue is reported in USD using scale-factor 1; in this case, there is no conflict. The second sub-query handles tuples for which revenue is reported in JPY, implying a scale-factor of 1000. Finally, the last sub-query considers the case where the currency is neither JPY nor USD, in which case only currency conversion is needed. Conversion among different currencies is aided by the ancillary data source r3 (a Web service) which provides currency conversion rates. This second query, when executed, returns the "correct" answer consisting only of the tuple <'NTT' 9 600 000>.

4 Conclusion

Together with our industry partners, we are currently deploying our technology in several experimental applications, an example of which is the area of financial analysis decision support (profit and loss analysis, and marketing intelligence). We have built several demonstrations which provide access to a number of on-line databases providing financial and company profiles. In addition, we also provide access to Web sites, which serve both as a primary source of information (for instance, sites reporting security prices on the various stock exchanges at regular intervals) or as ancillary data sources that are useful for realizing data transformations from one context to another (for instance, sites reporting currency exchange rates are used to support conversion between monetary amounts reported in different currencies).

References

[GBMS96] C. H. Goh, S. Bressan, S. E. Madnick, and M. Siegel. Context Interchange: Representing and Reasoning about Data Semantics in Heterogeneous Systems. Technical Report #3928, Sloan School of Management, MIT, 50 Memorial Drive, Cambridge MA 02139, October 1996.

[GMS94] C. H. Goh, S. Madnick, and M. Siegel. Context interchange: overcoming the challenges of large-scale interoperable database systems in a dynamic environment. In *Proceedings of the Third International Conference on Information and Knowledge Management*, pages 337–346, Gaithersburg, MD, Nov 29–Dec 1 1994.

[KK93] A. Kakas and F Kowalski, R and. Toni. Abductive logic programming. *Journal of Logic and Computation*, 2(6):719–770, 1993.

[KL89] M. Kifer and G. Lausen. F-Logic: a higher-order language for reasoning about objects, inheritence and scheme. In *Proc ACM SIGMOD*, pages 134–146, 1989.

[Qu96] J. Qu. Data wrapping on the world wide web. Technical Report CISL WP#96-05, Sloan School of Management, Massachusetts Institute of Technology, February 1996.

[SL90] A. Sheth and J. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3):183–236, 1990.

[SSR94] E. Sciore, M. Siegel, and A. Rosenthal. Using semantic values to facilitate interoperability among heterogeneous information systems. *ACM Transactions on Database Systems*, 19(2):254–290, June 1994.

[Wie92] G. Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer*, 25(3):38–49, March 1992.