Semantic Integration of Disparate Information Sources over the Internet Using Constraint Propagation

Stephane Bressan & Cheng Goh

Sloan WP# 3976 CISL WP# 97-10
August 1997

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

Semantic Integration of Disparate Information Sources over the Internet using Constraint Propagation

Stéphane Bressan

Sloan School of Management
Massachusetts Institute of Technology

steph@context.mit.edu

Cheng Goh

Dept. of Information Systems and Computer Science
National University of Singapore

gohch@iscs.nus.sg

Abstract

The Context Interchange Project is studying the semantic integration of disparate, i.e. distributed and heterogeneous, information sources. In this paper, we illustrate the principle of Context Mediation using examples from an application of our prototype. We outline the Context Mediation abductive framework and present the Context Mediation procedure. We show how this procedure is implemented using a constraint store and a constraint propagation model. We discuss the relationship between our approach and previous work on semantic query optimization.

1. Introduction

The Context Interchange Project at MIT [Goh94] is studying the semantic integration of disparate, i.e. distributed and heterogeneous, information sources. Like many other information integration projects (the SIMS project at ISI [Are92], the TSIMMIS project at Stanford [Gar95], the DISCO project at Bull-INRIA [Tom95], the Information Manifold project at AT&T [Lev95], the Garlic project at IBM [Pap96], the Infomaster project at Stanford [Dus97]) we have adopted the Mediation architecture outlined in Wiederhold's seminal paper [Wie92]. Figure 1 represents the architecture of the Context Mediation network. The central component of the Context Interchange network prototype (called MINT) [Bre97a,Bre97b] is the Mediator which provides the set of mediation capabilities: multi-source query processing (query planner/optimizer and executioner), and semantic conflict detection and resolution (Context mediator). The wrappers are gateways between the sources and the Mediator: they provide physical connectivity and an initial level of logical connectivity by providing a relational query interface to various types of sources including databases and semi-structured Web sites[1]. At the front end, user and application programming interfaces facilitate the integration of the integrated service into desktop applications (Web browsers, spreadsheets, personal databases, etc).

In section 2, we illustrate the principle of Context Mediation using examples from an application of our prototype [Bre97b]. In section 3, we outline the Context Mediation abductive framework and present the Context Mediation procedure. In section 4, we show how this procedure is implemented using a constraint store and a constraint propagation model [Jaf96]. We discuss the relationship between our approach and previous work on semantic query optimization. In concluding, we indicate our future work directions.

[1] It is out of the scope of this paper to explain how we construct relational interfaces to semi-structured Web sites. The reader can refer to [Bre97d,Bon97] for a discussion on this matter.

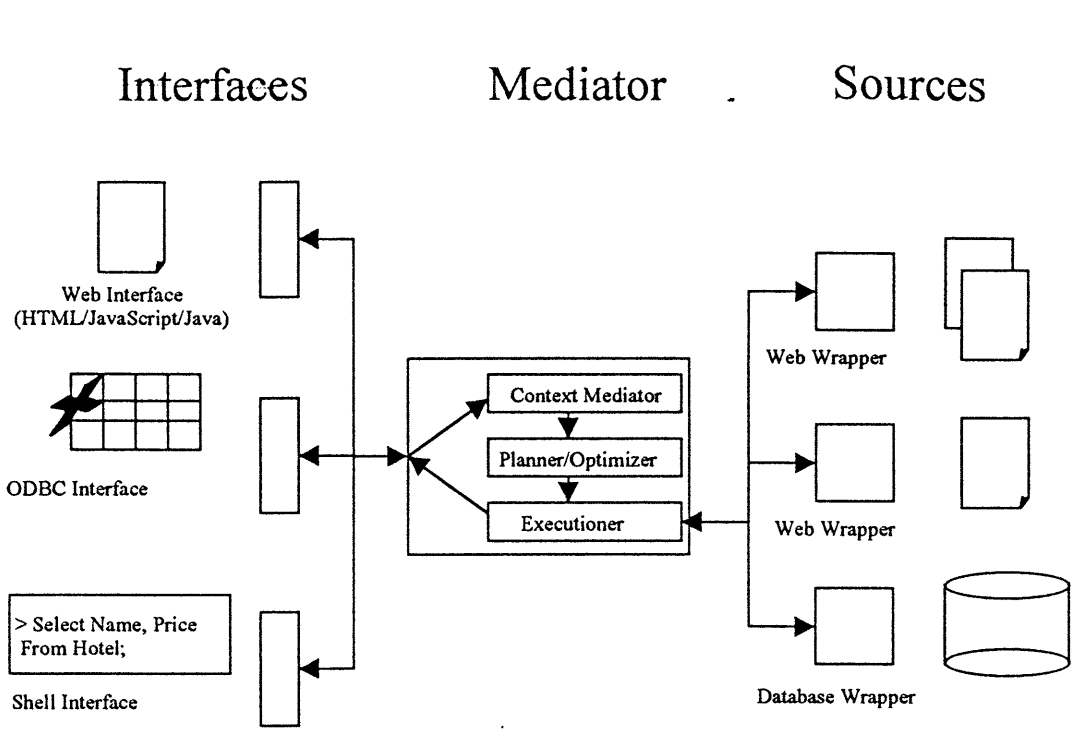

Figure 1: Architecture of the Context Interchange Network Prototype, MINT

2. Context Mediation

The application scenario considered here is the construction of a global, virtual hotel guide from some of the numerous on-line databases and Web sites that provide lodging information. A simple search with one of the commonly used search engines (e.g. Yahoo) retrieves hundreds of links to Web sites providing such information. In addition, hotel chains or travel agents have access to proprietary databases of hotels and lodging facilities. Name, location, and prices belong to the subset of data provided by most of these sources, and constitute the basic schema and search interface for our federated system. A sample of pages (input form, hit list, hotel description and pictures) from HOTELG, a prototypical Web-based hotel guide which is used as one of the data sources in our application, is depicted in Figure 2.

Let us now assume that we submit a query to the Context Mediation. Queries are expressed in plain SQL and evaluated against the federated schema exported by the Mediator. In our example, the part of the export schema with which we are interested consists of the relation "HOTEL." The attributes of the HOTEL relation are: "NAME", "COUNTRY", "STATE", CITY", "CURRENCY", "SINGLE", and "DOUBLE", which correspond to the name of the hotel, the country, state and city of location, currency in which prices are expressed, and the prices of single and double rooms, respectively.

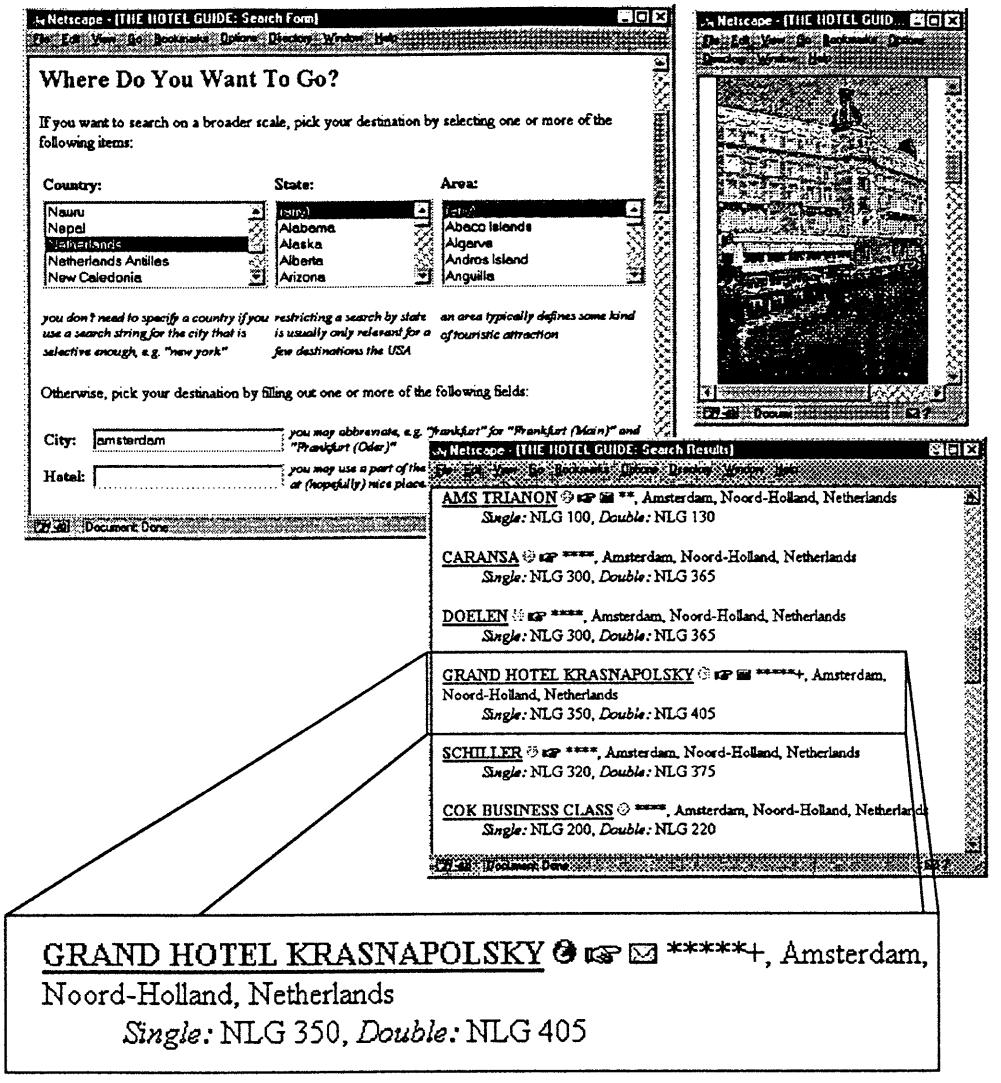

Figure 2: Sample Pages from the HOTELG Web source

Planning our next vacation to Senegal, we inquire about hotel rates in Dakar. We ask the following SQL query:

Select COUNTRY, NAME, DOUBLE From HOTEL Where CITY="Dakar";

The Mediator retrieves the following table of results:

hotel.Country	hotel.Name	hotel.Double
Senegal	NINA	208.00
Senegal	AACHA	11150.00
Senegal	AL AFIFA	74.00
Senegal	AL BARAKA	13000.00
Senegal	FARID	12500.00
Senegal	LE LAGON II	458.00
Senegal	NOVOTEL DAKAR	123.00
Senegal	SAINT-LOUIS SUN	19600.00

Upon first glance, the differences in the prices are quite striking. We might be tempted to select the cheapest hotel, "AL AFIFA", and to rule out the most expensive ones: "AACHA", "AL BARAKA", and "FARID". However, closer inspection reveals that because the observed prices are not all expressed in the same currency, our initial reaction is not well founded. Indeed more

careful consideration of the data reveals that hotel prices for the city of Dakar are expressed in three different currencies: French Francs ("FRF"), West African Francs ("XOF"), and United States Dollars ("USD") as shown by the results to the following query:

```
Select CURRENCY from HOTEL where City="Dakar" and COUNTRY="Senegal";
```

hotel.Currency
FRF
XOF
USD

The Context Interchange Mediator prototype, when requested, uses Context knowledge to automatically identify and resolve such semantic conflicts. Context knowledge consists of a domain model, a set of context definitions, sets of axioms associating the sources and respective source relations to the domain model and contexts, and conversion knowledge to resolve identified conflicts. The domain model is a collection of types corresponding to the entities subject to different interpretations or representations in the sources. For instance the domain model for international business used in the example comprises types for money amounts, dates, units of measurement, etc. The contexts are sets of axioms associated with both sources and receivers and define the preferred interpretation for the instances of the types in the domain model. For instance a context for a German user might specify that prices (which are monetary amounts) are both expected and expressed in German Marks. With Context Mediation, when framed in a German context, the query:

```
Select NAME, DOUBLE From HOTEL Where CITY="Dakar";
```

now returns a set of results where the currencies are all normalized to German Marks. The hotels whose prices once seemed the most extravagant are now the most affordable and vice versa:

hotel.Name	hotel.Double
NINA	62.40
AACHA	34.57
AL AFIFA	128.02
AL BARAKA	40.30
FARID	38.75
LE LAGON II	137.40
NOVOTEL DAKAR	212.79
SAINT-LOUIS SUN	60.76

The semantic heterogeneity among sources and between sources and receivers may require a deep reformulation of the query and not just, as might have been inferred from this first example, a simple mapping of results. For instance the query:

Select NAME, DOUBLE From HOTEL Where City="Dakar" and DOUBLE < 80;

returns disjoint sets of results with and without context mediation as illustrated using the German context in the tables below:

result without Context Mediation:

hotel.Name	hotel.Double
AL AFIFA	74.00

result with Context Mediation in the German context:

hotel.Name	hotel.Double

NINA	62.40
AACHA	34.57
AL BARAKA	40.30
FARID	38.75
SAINT-LOUIS SUN	60.76

3. The Mediation Abductive Procedure

Context Mediation is a view rewriting process. Unlike standard federated systems, the views in the Context Mediation system are not entirely statically defined. The part of the views that correspond to the resolution of semantic conflicts is dynamically determined through the comparison of source and receiver contexts. Because context definitions and the functions which implement conversions between contexts are kept independent of specific views, this information can be reused, introducing flexibility and scalability to the integration process.

When submitted to the Mediator, an SQL query is first compiled into a Datalog query representing the query and the relationship between the attributes in the query and instances of the attribute types in the domain model. This syntactic transformation is based on the axioms linking the relations exported by the sources to the domain model. The Context Mediator rewrites the query against the view that it dynamically composes. Namely, the query is augmented with additional operations or subqueries to additional datasources for data transformation and conflict resolution.

In the example above requiring a currency conversion, the system proposes the "OLSEN" Web site, which provides historical[2] data about exchange rates between major currencies. In the Context Mediation system, this Web site is treated as a data source that exports a relation "OLSEN" with the attributes "EXCHANGED", "EXPRESSED", "RATE", and "DATE", to represent the expressed currency, the exchanged currency, the exchange rate, and the date for which the rate is valid, respectively[3]. For instance, the following SQL query inquiring about hotels double rooms in the city of Cambridge (asked in a French context):

```
select Name, Double from hotel where City="Cambridge";
```

is rewritten into the Datalog query:

```
answer(V13, V12) :-
        hotelg(V13, V11, V10, "Cambridge", "FRF", V9, V12).
answer(V8, V7) :-
        hotelg(V8, V6, V5, "Cambridge", V4, 'V3, V2),
        olsen(V4, "FRF", V1, "7/05/1997"),
        V7 is V2 * V1.
```

The two rules correspond to the two possible cases: either the prices are already expressed in French Francs and do not need to be translated, or they are in a different currency and the conversion rate to French Francs needs to be applied.

[2] In the example the context mediator substitutes the date for the current date. Other applications such as international portfolio management, based on the same domain model, may need to access past exchange rates.

[3] As referenced in footnote 1, please refer to [Bre97d,Bon97] for information on the construction of a relational interface to semi-structured Web sites.

The domain model, the contexts, and all axioms and view definitions are expressed in the COINL language [Goh97]. COINL is a customized subset of F-Logic [Kif95]. COINL is a deductive and object oriented language. A COINL program constitutes a Horn Clause theory T. The Context Mediation of a query Q is defined as the abductive inference [Kak93] of the subqueries from the query Q and the theory T under some additional integrity knowledge about the sources which we discuss in the next section. The abductive framework for Context Mediation is defined by the following elements:

- the COINL program (domain model, context, conversions, etc) expressed as a theory T of Horn clauses;
- the set P of all predicate symbols used in T composed into the disjoint union of two sets Pc and Pa. Pc contains the predicates corresponding to context axioms and to views (e.g. HOTEL). Pa contains the predicates corresponding to the relations exported by the sources, the ancillary relations used to resolve the conflicts (e.g. OLSEN), and other predicates evaluable at execution time (e.g. arithmetic). Pa is called the set of abducible predicates. We say that a literal is abducible if the literal's predicate symbol is in Pa.
- a set of integrity constraints IC describing integrity knowledge about the sources and ancillary relations.

Q' is defined to be inferred from Q and T by abduction under IC if and only if:

- Q' is a conjunction of abducible literals.
- Q is a logical consequence of T and Q'
- T, IC, and Q' are consistent;

Because we do not allow recursive view definitions, this inference can be implemented by a slightly modified SLD-resolution of the query Q against the program T where the resolution of the abducible literals is indefinitely delayed. Intuitively, the algorithm can be described as the top-down resolution of the query against the view definitions and the context knowledge where the evaluation of subqueries to remote sources is left to a subsequent phase.

We have implemented the Mediator and the abduction procedure in the ECLiPSe parallel constraint logic programming environment [ECL95a]. The Prolog program below sketches the algorithm's steps. The set of abducible literals, which is set aside during the development of one branch of the proof tree, is implemented as a constraint store by coroutining.

```
abduct(Query, SubQueries) :-
      setof(Store, abduct_and_collect(Query, Store), SubQueries).

abduct_and_collect(Query, Store):-
      sub_abduct(Query),
      collect(Store).

sub_abduct([]).
sub_abduct([QueryHead|QueryTail]):-
      view(QueryHead, ViewTail),
      append(ViewTail, QueryTail, NewQuery),
      sub_abduct(NewQuery).
sub_abduct([QueryHead|QueryTail]):-
      abducible(QueryHead),
      post(QueryHead),
      NewQuery = QueryTail,
      sub_abduct(NewQuery).
```

The procedure sub_abduct/1 implements the modified conventional SLD resolution. The post/1 procedure posts abducible literals to the store. As we will discuss in the next section, the store's consistency is automatically maintained by the propagation of constraints expressed as Constraint Handling Rules [Ecl95b]. For each branch terminating without a failure, the literals in the store are collected (procedure collect/1) to form one subquery.

4. Semantic Query Optimization and Constraint Propagation

The store's consistency is maintained by means of a set of Constraint Handling Rules (CHRs) [Ecl95b]. The rules correspond to generic inequality and disequality constraints, to integrity constraints, and, as required by the application, to specific constraints such as linear equations.

The addition of a new literal to the store is propagated by the CHR engine. The engine adds or removes literals from the store, and possibly binds variables. The variable bindings are propagated back into the resolution branch. If the store is inconsistent, an inconsistency is eventually detected and the resolution branch is abandoned. Abandoned branches correspond to inconsistent subqueries, i.e. those queries, which would return an empty result if evaluated. Pragmatically, the completeness of the consistency test is not essential to the procedure since it mainly implements an optimization, which rules out subqueries whose subsequent evaluation would return an empty result. As we will see, the system can already take advantage of the addition of new literals corresponding to additional selections in the subsequent evaluation of the subquery.

We need now to illustrate the compilation of the integrity constraints into CHRs. An integrity constraint is a formula of the form ¬(l1, ...,ln), i.e. a Horn clause containing only negative literals. The li's are abducible litterals. For each literal lj corresponding to an equality or disequality the IC is compiled into a CHR of the form:

```
l1,...,lj-1,lj+1,ln ==> ~~lj~~
```

where ~~lj~~ stands for the positive counterpart of the negation of lj.

For instance, a functional dependency expressed by the integrity constraint:

```
¬ (p(X,Y), p(X, Z), Z\=X)
```

is compiled into the CHR:

```
p(X,Y), p(X,Z) ==> X=Y
```

which will propagate the equality as soon as the LHS of the CHR matches a subset of the store.

For example, for the hotel guide introduced in Section 2, we might wish to express the integrity knowledge that the HOTELG data source *reports all prices for all United States hotels in US Dollars.* This constraint is expressed by a formula of the form:

```
HOTELG(X, Y, Z, T, U, V, W), Y = "USA" -> U = "USD"
```

The constraint is compiled into two CHRs:

```
HOTELG(X, Y, Z, T, U, V, W), Y = "USA" ==> U = "USD"
HOTELG(X, Y, Z, T, U, V, W), Y \= "USD" ==> U \= "USA"
```

If we now ask the following query in a French context (prices are expected and expressed in French Francs):

```
select Name, Double from hotel where City="Cambridge" and
                        Country="U.S.A.";
```

the mediated query in Datalog is:

```
answer(V6, V5) :-
        hotelg(V6, "U.S.A.", V4, "Cambridge", "USD", V3, V2),
        olsen("USD", "FRF, V1, "7/05/1997"),
        V5 is V2 * V1.
```

Comparing this result with the example of the previous section, we observe that the condition on the country (here incorporated in the query itself) has triggered the use of the first CHR and has lead to the realization that a conversion from US Dollars to French Francs is needed. Similarly, were the same query asked in an American context (prices are expected and expressed in US Dollars), the system would identify the fact that no currency conversion is necessary. The mediated query in Datalog is:

```
answer(V4, V3) :-
        hotelg(V4, "U.S.A.", V2, "Cambridge", "USD", V1, V3).
```

Semantic Query Optimization, as described in [CGM90], is the process of increasing the potential for an efficient evaluation of database queries using semantic knowledge expressed by integrity constraints. For example, in the simplest case, a query may be answered without accessing a particular datasource if it is known, a priori, that all data at that source is inconsistent with the constraints (i.e. the answer to the query is the empty set). Integrity knowledge can also be used to augment the query with additional selections, which may restrict the amount of data retrieved from individual relations or enable the use of keys and indices. These examples correspond in the classification of King [Kin81] to the cases of "detection of unsatisfiable conditions", "scan reduction", and "index introduction," respectively. These are the exact cases treated by our Context Mediation procedure.

In addition, King identifies two additional cases: "join elimination" and "join introduction". These cases correspond to the introduction or elimination of literals into the subqueries. To clarify, the same integrity constraint can lead to either the introduction or the elimination of a given literal. Therefore, additional information must be used to define the heuristic distinguishing between introduction, elimination, or neither. Such a heuristic can only be based on a cost model. We have not yet been able to define a satisfying cost model for our system given that we have no control over the remote sources due to the principle of non-intrusiveness.

Finally, we remark that our method is comparable to the method of Chakravarthy et al [CGM90] for compiling integrity constraints into residues attached to the rules (view definitions) of a deductive database. We have found, however, that the ad-hoc nature of the queries in our applications does not require a pre-compiled approach. Leveraging the CHR engine we also benefit by the possibility of extending the query language [Kan90] to application specific constraint domains and seamlessly extend the semantic query optimization capabilities to these domains.

5. Conclusion

Based upon examples drawn from a working prototype, we have presented the main features of the Context Interchange system. We have discussed and illustrated the rationale of the Context Mediation abductive procedure and outlined its implementation based on the relationship between constraint propagation and integrity constraint consistency checking. The basic ideas behind this approach can be traced back to the work of Finger and Genesereth [Fin85] on abduction and

constraint propagation. More recently, Wetzel, Kowalski, and Toni have proposed the use of a similar framework as a new programming language paradigm.

From the standpoint of integrating disparate information systems, we need now to investigate the usefulness of the generalization of semantic query optimization to application specific domains [Wal96]. Working with our industrial partners, we plan to experiment with finite domains and rational linear equations and disequations solvers in logistic and accounting applications, respectively.

As an extension of our work, we have also identified the potential for taking advantage of "join elimination" and "join introduction" to recover from the impossibility of executing subqueries due to certain capabilities restrictions of remote sources [Pap95] (in particular Web sources). Indeed, as certain sources allow only a limited set of query patterns (e.g. the input of a city is required to search the hotel guide), integrity constraints can be used to introduce additional literals and to enable subquery evaluation. For example, inclusion dependencies relating the cities in a source to a list of cities in another source can be used to generate the necessary inputs.

More generally, we believe that propagating integrity constraint knowledge as constraints in query evaluation to disparate sources can simulate user's strategies for gathering information from the global information infrastructure and lead to the construction of intelligent and efficient information based decision support systems.

References

[Are92] Arens, Y. and Knobloch, C. *"Planning and reformulating queries for semantically-modeled multidatabase"*. Proc. of the Intl. Conf. on Information and Knowledge Management. 1992.

[Bon97] Bonnet, Ph., and Bressan, S. *"Extraction and Integration of Data from Semi-structured Documents into Business Applications"* Submitted.1997.

[Bre97a] 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.

[Bre97b] Bressan, S. et al. "The Context Interchange Mediator Prototype". Proc of ACM-SIGMOD. (see also http://context.mit.edu/demos/sigmod). 1997.

[Bre97c] 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.

[Bre97d] Bressan, S., and Lee, T. *"Information Brokering on the World Wide Web"*. To appear in the Proc. of the WebNet World Conf. 1997.

[Cha90] Chakravarthy, U., Grant, J., and Minker, J. *"Logic-Based Approach to Semantic Query Optimization"*. ACM Trans. on Database Sys. 15:2. 1990.

[Dus97] Duschka, O., and Genesereth, M. *"Query Planning in Infomaster"*. http://infomaster.stanford.edu. 1997.

[Ecl95a] ECRC. *"Eclipse Manual"*. ECRC. 1995.

[Ecl95b] ECRC. *"Eclipse Extensions Manual"*. ECRC. 1995.

[Fin85] Finger, J., and Genesereth, M. *"Residue: A Deductive approach to design synthesis"*. Stanford University TR-CS-8.1985.

[Gar95] Garcia-Molina, H. *"The TSIMMIS Approach to Mediation: Data Models and Languages"*. Proc. of the Conf. on Next Generation Information Technologies and Systems. 1995.

[Goh94] 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.
[Goh97] 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.
[Jaf96] Jaffar, J., and Maher, M. *"Constraint Logic Programming: A Survey"*. J. of Logic Programming. 1996.
[Kak93] Kakas, A., Kowalski, R., and Toni, F. *"Abductive Logic Programming"*. Journal of Logic and Computation 6:2. 1993.
[Kan90] Kanellakis, P., Kuper, G., and Revesz, P. *"Constraint Query Languages"*. To appear in J. of Computer and System Sciences (a preliminary version appeared in Proc. of 9th PODS 1990.
[Kif95] Kifer, M., Lausen, G., and Wu, J. *"Logical foundations of object-oriented and frame-based languages"*. J. of the ACM-SIGMOD. 1995.
[Kin81] King, J. *"Query Optimization by Semantic Reasonning"*. Stanford University, Ph.D. Thesis. 1981.
[Lev95] Levy, A., Srivastava, D., and Kirk, T. *"Data Model and Query Evaluation in Global Information Systems"*. J. of Intelligent Information Systems. 1995.
[Pap95] 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.
[Pap96] 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.
[Tom95] 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.
[Wal96] Wallace, M. *"Practical Applications of Constraint Programming"*. Constraints, An International Journal 1. 1996.
[Wet96] Wetzel, G., Kowalski, R., and Toni, F. *"Procalog: Programming with Constraints and Abducibles in Logic"*. JICSLP Poster session. 1996.
[Wie92] Wiederhold, G. *"Mediation in the Architecture of Future Information Systems"*. Computer, 23(3). 1992.