Contexts in Agent-Mediated Electronic Commerce

Chiara Ghidini
Dept. of Computing and Mathematics
Manchester Metropolitan University
Manchester M1 5GD, United Kingdom
C.Ghidini@doc.mmu.ac.uk

Luciano Serafini
ITC-IRST, 38050 Povo, Trento, Italy
serafini@itc.it

Abstract

In agent-mediated electronic commerce, agents need to exchange information with other agents and to integrate the obtained information with their own data. Integration is a very complex task as: information is *distributed* among different agents; each agent *autonomously* represents and manages its own information; information might be *partial*, as agent cannot wait to have complete information before acting; finally information is *redundant*, as the same information might be represented by two different agents. Our goal is to provide a formal semantics for information integration able to cope with distributed, autonomous, partial, and redundant information. Such a semantics is based on the intuition that agents' databases can be thought as contexts, each context representing the partial view of an agent on the common world. In the paper we introduce an example from an electronic commerce scenario and we show that a context based semantics for information integration allows us to cope with some of the problems of information integration emphasized in the example.

Introduction

In agent-mediated electronic commerce each agent is associated with a system for the management of the information. This information constitutes agents *goals*, *plans*, and *beliefs* about the state of the market. An agent plans and performs actions, such as negotiation, evaluation of different offers, contract stipulation, etc., on the basis of the information available in its system. In many cases however, an agent doesn't have enough information for pursuing its goals and, in order to properly plan its actions, it needs to collect suitable extra information from other agents. In defining a system for the management of the agent's information, we have therefore to consider at least two aspects. The first concerns the internal structure of such a system. This includes how beliefs, goals, plans, etc., are represented and how to reason about them. The second aspect concerns how the information, obtained by communicating with the other agents, is integrated in the agent's information. In this paper we address the problem of the integration of information that different agents have. We don't consider here representational issues such as agents' belief, goals, and so on[1]. For this reason we introduce a simplifying hypothesis on the structure of the information system of each agent: we suppose that agents represent information by a relational database.

Focusing on the problem of exchanging and integrating information, a set of agents can be abstracted to a set of databases able to communicate via some agent communication mechanism. These databases are *distributed*, *partial*, *autonomous*, and *redundant*. Distribution means that databases of different agents are different systems, each of them containing a specific piece of information. Partiality means that the information contained in a database may be incomplete. Autonomy means that the database of each agent is autonomous regarding the design, the execution, and the communication with the other agents. Therefore different databases may adopt different conceptual schemata (including domain, relations, naming conventions, ...). Redundancy means that the same piece of information may be represented, possibly from different perspectives, in the databases of different agents. Redundancy not only means that information is duplicated, but also that the information of two databases might be related. Redundancy is what makes communication possible. Indeed communication (as intended here) allows information to be duplicated from an agent to another. This is possible only if some data can be stored in the database of the two agents.

Distribution, partiality, autonomy, and redundancy generate many problems in integrating the information contained in different databases. Important problems are: semantic heterogeneity, interschema dependencies, query distribution , local control over data and processing, and transparency. The definition of a formal semantics for information integration able to cope with these problems is a key point to understand, specify, and verify the behavior of a multi-agent system for electronic commerce.

Several approaches have been proposed in the past.

[1]We have addressed representational issues, such as multi agent beliefs, in some previous paper (see for instance (Giunchiglia & Serafini 1994)). These approaches are homogeneous with the formalism presented in this paper and can be easily integrated in it.

An incomplete list is (Catarci & Lenzerini 1993; Mylopoulos & Motschnig-Pitrik 1995; Subrahmanian 1994; Guha 1990; McCarthy & Buvač 1998). However all these approaches fail in representing the issues listed above in a uniform way. This failure is due, from our perspective, to the fact that the semantics of the agents' databases are built by filtering the information contained in a complete description of the real world. However a complete description of the real world is hardly to be available, especially in the case of a multi-agent scenario. In most of the cases, indeed, the agents' databases have been developed independently and each of them contains a partial and subjective description (view) of the real world. Therefore the semantics for information integration must be defined in terms of these views of the real world.

The semantics proposed in this paper is an extension to first order languages of a the semantics of contexts proposed in (Giunchiglia & Ghidini 1998; Ghidini & Serafini 1998a), called *Local Model Semantics* (LMS hereafter). It is based on the intuition that the database of an agent can be though as a context, each context having its own (local) language and semantics. The exchange and integration of information between databases is modelled in terms of (compatibility) relations between contexts.

The paper is structured as follows. In the next section we introduce an example which emphasizes critical problems of information integration. Then we briefly introduce the concept of integration information schema and we define LMS for information integration. Finally we formalize the example via LMS and we make some concluding remarks.

An Explanatory Example

Let I be a company which produces and sells assessments for Italian cars, and E be a company which does the same for European sport cars. The assessment process is the same for both companies and consists of 10 tests (e.g. for comfort, brakes, consume, ...). For each test a car is assigned a rating from 1 to 10. I assigns a final score from A to F computed as follows: a car is assigned an A if its total evaluation after the 10 tests is less than $\frac{100}{6}$, a B if it is between $\frac{100}{6}$ and $2 * \frac{100}{6}$, and so on. E instead assigns a final score from 0 to 10 obtained by dividing the total valuation by 10 and rounding to the nearest half point. Figure 1 compares the two scales. Suppose that, in assessing to *car1*, I

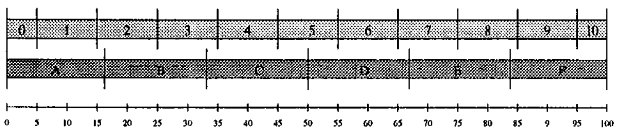


Figure 1: Comparison of the different scales

assigns *car1* a total evaluation of 80 after the first 9 tests, but for some reason the tenth test cannot be done. Since I doesn't want to loose the data of the previous 9 tests, a partial information is included in the assessment database of I and *car1* will be assigned a partial evaluation which is either E or F. I and E decide to collaborate by exchanging the data about common products (i.e. Italian sports cars). However they are not completely cooperative. Indeed I decides to communicate only complete information (i.e. no information about partial tests are communicated) and E decides not to communicate the assessments of a specific sport car, say *Ferrari F40*, because it is to critical. A formal model of this example must address the following aspects:

Semantic heterogeneity The two scales 0–10 and A–F constitute an heterogeneous measure of the same aspects;

Different domains The domain of European sport cars is different from that of Italian cars;

Domain overlapping European sport cars and Italian cars overlap on Italian sport cars;

Partiality The incomplete assessment of *car1* generates partial information;

Autonomy of communication Each company decides to communicate only a specific set of data.

LMS for Information Integration

The formalization of information integration is done in two steps. In the first step we define an *information integration schema* which describes the structure of the database of each single agent and how the information of different agents is integrated, in the second step we define an *information integration state* as a formal model of an information integration schema.

Let I be a (at most) countable set of indexes, each denoting an agent (or equivalently its database). The first component of an information integration schema are the schemata of the agents' databases. Since agents represent information via relational databases, an information integration schema is build on top of a family $\{S_i\}_{i \in I}$ (hereafter $\{S_i\}$) of relational schemata (Abitebul, Hull, & Vianu 1995).

The second component of an information integration schema represents how the databases of the different agents are integrated. Since information is distributed, we cannot assume the existence of a common data structure, shared by different agents. Therefore agents can communicate only via query answering. Since information is heterogeneous, if an agent, let's say the agent E in our example, wants to collect information from I about the set of individuals which satisfy the property of being an Italians sport car, then it must perform the following operations:

1. rewriting its own query $\psi(x)$ (which means "is x an Italians sport car?") in the query $\psi(x)$ in the language of I;
2. mapping back the answer of I (which is a set of cars in the domain of I) into a set of cars in its own domain.

The specification of query rewriting schemata and answer rewriting schemata are obtained introducing the definitions of *view constraints* and *domain constraints* respectively.

Definition 1 (View Constraint) *Let S_i and S_j be two database schemata. A view constraint from S_i to S_j is an expression $i:\phi(x_1,\ldots,x_n) \rightarrow j:\psi(x_1,\ldots,x_n)$, where $\phi(x_1,\ldots,x_n)$ and $\psi(x_1,\ldots,x_n)$ are formulae (queries) in the language of S_i and S_j, respectively.*

Intuitively $i : \phi(x_1,\ldots,x_n) \rightarrow j : \psi(x_1,\ldots,x_n)$ means that the agent j can ask the agent i the sub-query $\phi(x_1,\ldots,x_n)$ in order to answer the query $\psi(x_1,\ldots,x_n)$.

Definition 2 (Domain Constraint) *Let S_i and S_j be two database schemata. A domain constraint from S_i to S_j is an expression of the form $\mathsf{T}^{i:A}_{j:B}$ or $\mathsf{S}^{i:A}_{j:B}$ where A and B are formulae with one free variable of S_i and S_j respectively.*

Intuitively $\mathsf{T}^{i:A}_{j:B}$ means that, from the point of view of j, for any object of in the domain of i, denoted by $\mathbf{dom}_i$, which is in the answer set of the query A there is a corresponding object in its own domain $\mathbf{dom}_j$ which is in the answer set of the query B. Conversely $\mathsf{S}^{i:A}_{j:B}$ means that from the point of view of j, for any object in $\mathbf{dom}_j$ which is in the answer set of B, there is a corresponding object in $\mathbf{dom}_i$ which is in the answer set of A.

An *interschema constraint* IC_{ij} from S_i to S_j is a set of view and domain constraints from S_i to S_j

Definition 3 (Information Integration Schema) *An Information Integration Schema on I is a pair $IIS = \langle\{S_i\},\{IC_{ij}\}\rangle$ where, for each $i,j \in I$ with $i \neq j$, S_i is a database schema and IC_{ij} is an interschema constraint from S_i to S_j.*

An information integration schema describes a class of possible combinations of agents' databases, i.e. those combinations which satisfy the interschema constraints. The combination of the agents' databases at a given instant point is called *information integration state.*

An information integration state is defined by formalizing each agent's database as a context and by taking the perspective described in (Giunchiglia & Ghidini 1998) for the semantics of contextual reasoning. The formal semantics associated to each database i represents the description of the current state of the world from i-th partial point of view. Therefore the formal semantics of the information integration schema $\langle\{S_i\},\{IC_{ij}\}\rangle$ contains a set $\{\mathrm{DB}_i\}$ of databases, each DB_i being a partial database on the schema S_i. According to the perspective described in (Giunchiglia & Ghidini 1998), databases may have distinct domains. In order to propagate information about related objects we introduce a family of *domain relations* between different databases' domains. Formally a domain relation r_{ij} from $\mathbf{dom}_i$ to $\mathbf{dom}_j$ is a subset of $\mathbf{dom}_i \times \mathbf{dom}_j$. Intuitively r_{ij} represents the capability of agent j to map the objects of $\mathbf{dom}_i$ in its domain $\mathbf{dom}_j$. In particular $\langle d, d'\rangle \in r_{ij}$ means that, from the point of view of j, d in $\mathbf{dom}_i$ is the representation of d' in its own domain.

A formal semantics for information integration is composed of a set of databases (see, for instance, (Abitebul, Hull, & Vianu 1995)) and set of domain relations from the schema of a component to that of the others. Domain constraints imply that only certain domain relations are accepted. In particular $\mathsf{T}^{i:A}_{j:B}$ implies that the relation r_{ij}, restricted to the answer sets of $i : A$ and $j : B$, must be total, whereas $\mathsf{S}^{i:A}_{j:B}$ means that r_{ij}, restricted to the answer sets of $i : A$ and $j : B$, must be surjective. View constraints imply that only certain combinations of databases are admitted. In particular $i : \phi(x_1,\ldots,x_n) \rightarrow j : \psi(x_1,\ldots,x_n)$ implies that DB_i and DB_j are such that the image of the query $\phi(x_1,\ldots,x_n)$ in DB_i is mapped, via r_{ij}, into a subset of the query $\psi(x_1,\ldots,x_n)$ in DB_j. These properties are formalized in (Ghidini & Serafini 1998a; 1998b) by the notions of satisfiability of domain constraints and view constraints, respectively.

Definition 4 (Information Integration State) *Let $\{\mathrm{DB}_i\}$ be a set of databases, each DB_i being a database on S_i, and $\{r_{ij}\}$ be a family of domain relations. An information integration state on the information integration schema $\langle\{S_i\},\{IC_{ij}\}\rangle$ is a pair $iis = \langle\{\mathrm{DB}_i\},\{r_{ij}\}\rangle$ such that for all $i,j \in I$, $\mathrm{DB}_i, \mathrm{DB}_j$, and r_{ij} satisfy the domain and view constraints in IC_{ij}.*

The interschema constraints contained in an information integration schema *IIS* imply that a certain fact ϕ in a database i, denoted with $i:\phi$, is a consequence of a set of facts Γ in, possibly distinct, databases. This relation is crucial as it allows to understand how information propagates through databases independently from the specific information state and is formalized in (Ghidini & Serafini 1998a; 1998b) by the notion of *logical consequence* $\Gamma \models_{IIS} i:\phi$.

Modeling the Example

In our example two agents, I and E contain two databases with local schemata S_I and S_E respectively.

Local Schemata Both S_I and S_E contain an attribute *value* (for evaluation values), ranging over $\{A,\ldots,F\}$ in S_I and over $\{1,\ldots,10\}$ in S_E, and an attribute *car* (for cars), ranging over the set of Italian cars in S_I and over the set of European sport cars in S_E. S_I and S_E contain a predicate $eval(x,y)$, of sort $\langle car, value\rangle$, meaning that the final score of car x is y; S_I contains a predicate *sport-car*(x), of sort *car*, meaning that x is a sport car; S_E contains a predicate *it-car*(x), of sort *car* meaning that x is an Italian car.

Interschema Constraints We remind that domain constraints from i to j represent the capability of j to map in its domain the answers of queries submitted to i. The complete cooperation of I in answering the queries submitted by E, corresponds to the fact that any sport car in $\mathbf{dom}_\mathsf{I}$ is translated in an Italian car in $\mathbf{dom}_\mathsf{E}$ and vice versa. This is represented by the domain constraints:

$$\mathsf{T}^{\mathsf{I}:sport-car(x)}_{\mathsf{E}:it-car(x)} \qquad \mathsf{S}^{\mathsf{I}:sport-car(x)}_{\mathsf{E}:it-car(x)} \tag{1}$$

In our example we must take into account that E is not completely cooperative as it refuses to give information about the car Ferrari *F40*. This means that, in answering a query submitted by I to E, any Italian car in $\mathbf{dom}_\mathsf{E}$ can be translated into a sport car in $\mathbf{dom}_\mathsf{I}$ and vice versa, apart from Ferrari *F40*. This is represented by the domain constraint

$$\mathsf{T}^{\mathsf{E}:it-car(x)\wedge x\neq F40}_{\mathsf{I}:sport-car(x)} \qquad \mathsf{S}^{\mathsf{E}:it-car(x)\wedge x\neq F40}_{\mathsf{I}:sport-car(x)} \tag{2}$$

Both companies are completely cooperative w.r.t the domains of the attribute *value*, that is I is able to translate any evaluation A–F of the domain of E in 1–10, and vice versa. This is represented by the following domain constraints:

$$\mathsf{T}^{\mathsf{I}:value}_{\mathsf{E}:value} \qquad \mathsf{S}^{\mathsf{I}:value}_{\mathsf{E}:value} \qquad \mathsf{T}^{\mathsf{E}:value}_{\mathsf{I}:value} \qquad \mathsf{S}^{\mathsf{E}:value}_{\mathsf{I}:value} \tag{3}$$

Let's consider view constraints. Both companies agree on cars names. E.g., the intended meaning of "car2" in the database of both companies is a unique car whose name is car2. This is represented by the view constraints:

$$\mathsf{I}: x = c \rightarrow \mathsf{E}: x = c \qquad \mathsf{E}: x = c \rightarrow \mathsf{I}: x = c \tag{4}$$

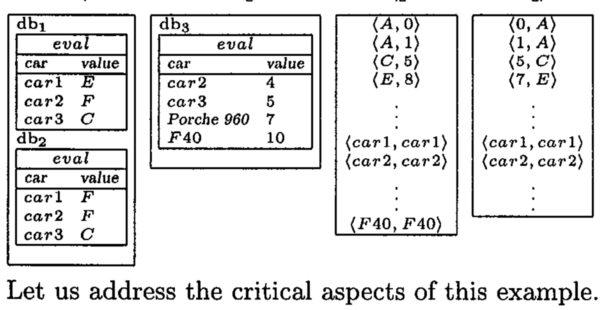
for any car name c which is in the language of I and E. Evaluation transformation is formalized by two sets of view constraints that reflect the comparison between the two different scales in Figure 1:

$$\begin{array}{ll} \mathsf{I}:x=A \rightarrow \mathsf{E}:x=0 \vee x=1 \vee x=2 & \mathsf{E}:x=0 \rightarrow \mathsf{I}:x=A \\ \mathsf{I}:x=B \rightarrow \mathsf{E}:x=2 \vee x=3 & \mathsf{E}:x=1 \rightarrow \mathsf{I}:x=A \\ \vdots & \vdots \\ \mathsf{I}:x=F \rightarrow \mathsf{E}:x=8 \vee x=9 \vee x=10 & \mathsf{E}:x=10 \rightarrow \mathsf{I}:x=F \end{array} \tag{5}$$

Finally, the intended meaning of the predicate $eval(x,y)$ in both databases coincides. This is formalized by the view constraints:

$$\begin{array}{l} \mathsf{I}: eval(x,y) \rightarrow \mathsf{E}: eval(x,y) \\ \mathsf{E}: eval(x,y) \rightarrow \mathsf{I}: eval(x,y) \end{array} \tag{6}$$

The information integration schema *IIS* for this example is composed by S_I, S_E, and the domain constraints and view constraints defined above. An example of information integration state *iis* on the schema *IIS* is:



DB$_\mathsf{I}$

db$_1$

eval	
car	value
*car*1	*E*
*car*2	*F*
*car*3	*C*

db$_2$

eval	
car	value
*car*1	*F*
*car*2	*F*
*car*3	*C*

DB$_\mathsf{E}$

db$_3$

eval	
car	value
*car*2	4
*car*3	5
Porche 960	7
*F*40	10

r_IE	r_EI
$\langle A, 0\rangle$	$\langle 0, A\rangle$
$\langle A, 1\rangle$	$\langle 1, A\rangle$
$\langle C, 5\rangle$	$\langle 5, C\rangle$
$\langle E, 8\rangle$	$\langle 7, E\rangle$
$\vdots$	$\vdots$
$\langle car1, car1\rangle$	$\langle car1, car1\rangle$
$\langle car2, car2\rangle$	$\langle car2, car2\rangle$
$\vdots$	$\vdots$
$\langle F40, F40\rangle$	

Let us address the critical aspects of this example.

Semantic heterogeneity View constraints (5) allow to relate the (heterogeneous) values in 0–10 and in A–F, influencing the definition of domain relations r_IE and r_EI. This fact allows I and E to exchange heterogeneous data about cars, as we show in the following. In the information integration state *iis* depicted above DB$_\mathsf{E}$ satisfies $eval(car3, 5)$. By view constraint (6), and by $\langle 5, C\rangle \in r_\mathsf{EI}$ it follows that DB$_\mathsf{I}$ satisfies $eval(car3, C)$. Another information integration state on the same schema is obtainable from *iis* by replacing $\langle 5, C\rangle$ with $\langle 5, D\rangle$ in r_EI (this is still a domain relation which satisfies view constraints (5)). Again view constraint (6) forces DB$_\mathsf{I}$ to satisfy $eval(car3, D)$. However in order to satisfy view constraint (5) and domain constraint $\mathsf{T}^{\mathsf{E}:value}_{\mathsf{I}:value}$ either $\langle 5, C\rangle \in r_\mathsf{EI}$ or $\langle 5, D\rangle \in r_\mathsf{EI}$. This means that, for any information integration state on the schema *IIS*, if DB$_\mathsf{E}$ satisfies $eval(car3, 5)$ then DB$_\mathsf{I}$ satisfies $eval(car3, C) \vee eval(car3, D)$. The above observations are summarized by the following properties of the logical consequence of *IIS*.

$$\mathsf{E}: eval(car3, 5) \not\models_{IIS} \mathsf{I}: eval(car3, C) \tag{7}$$
$$\mathsf{E}: eval(car3, 5) \not\models_{IIS} \mathsf{I}: eval(car3, D) \tag{8}$$
$$\mathsf{E}: eval(car3, 5) \models_{IIS} \mathsf{I}: eval(car3, C) \vee eval(car3, D) \tag{9}$$

The properties of $\models_{IIS}$ express that a one to one translation between rates doesn't exist because of the semantic heterogeneity between the two scales. In particular Equation (7 and Equation (8) formalize that we cannot translate the rate 5 to a unique value (C or D) because of the fact that 5 might be obtained rounding off a valuation between 4.5 and 5, or by rounding off a valuation between 5 and 5.5. However Equation (9) enable us to infer the partial information that car3's final score in the second scale is either C or D from the fact that car3's final score in the first scale is 5.

Different domains The domains of I and E contain different objects. For instance the domain of E contains the car *Porche 960* which is not an Italian car and it is not contained in the domain of I.

Domain overlapping The overlapping on Italian sport cars is formalized by domain constraints (1) and (2).

Partiality The incomplete assessment of *car1* is represented in *iis* by the fact that DB$_\mathsf{I}$ satisfies $eval(car1, E) \vee eval(car1, F)$ but it doesn't satisfy neither $eval(car1, E)$ nor $eval(car1, F)$.

Autonomy of communication I doesn't communicate to E any partial information about final scores. For instance, the partial rating of *car*1 in I does not entail any rating (even partial) in E, i.e.

$$\mathsf{I}: eval(car1, E) \vee eval(car1, F) \not\models_{IIS} \mathsf{E}: \exists x. eval(car1, x)$$

despite the fact that each disjunct entails a rating in E, i.e.

$$\begin{array}{l} \mathsf{I}: eval(car1, E) \models_{IIS} \mathsf{E}: eval(car1, 7) \vee eval(car1, 8) \\ \mathsf{I}: eval(car1, F) \models_{IIS} \mathsf{E}: eval(car1, 9) \vee eval(car1, 10) \end{array}$$

E doesn't send to I any data about Ferrari *F40*, i.e. for any evaluation X in 1–10

$$\mathsf{E}: eval(F40, X) \not\models_{IIS} \mathsf{I}: \exists x.eval(F40, x)$$

This follows from the fact that the domain relation r_{EI} might not associate any element to $F40$ (see domain constraint (2)). Non cooperativeness of E does not prevent I to be cooperative. Indeed for any evaluation X in A–F

$$\mathsf{I}: eval(F40, X) \models_{IIS} \mathsf{E}: \exists x.eval(F40, x)$$

Conclusions

In this paper we have described a formal semantics, called Local Models Semantics (LMS), for the integration of information of different agents. We have provided an example from the electronic commerce scenario involving some of the challenging aspects of information integration and we have described how LMS formalizes such aspects. In (Ghidini & Serafini 1998b) other examples from the electronic commerce scenario are formalized using LMS, whereas in (Ghidini & Serafini 1998a) a sound and complete calculus for LMS is provided.

References

Abitebul, S.; Hull, R.; and Vianu, V. 1995. *Foundation of Databases*. Addison-Wesley.

Catarci, T., and Lenzerini, M. 1993. Representing and using interschema knowledge in cooperative information systems. *International Journal of Intelligent and Cooperative Information Systems* 2(4).

Ghidini, C., and Serafini, L. 1998a. Distributed First Order Logics. In *2nd Int. Workshop on Frontiers of Combining Systems (FroCoS'98)*. To appear.

Ghidini, C., and Serafini, L. 1998b. Information Integration for Electronic Commerce. In *Workshop on Agent Mediated Electronic Trading (AMET'98)*, volume 1571 of *LNAI*. Springer Verlag.

Giunchiglia, F., and Ghidini, C. 1998. Local Models Semantics, or Contextual Reasoning = Locality + Compatibility. In *6th Int. Conference on Principles of Knowledge Representation and Reasoning (KR'98)*. Morgan Kaufmann.

Giunchiglia, F., and Serafini, L. 1994. Multilanguage hierarchical logics (or: how we can do without modal logics). *Artificial Intelligence* 65.

Guha, R. 1990. Microtheories and contexts in cyc. Technical Report ACT-CYC-129-90, MCC, Austin, Texas.

McCarthy, J., and Buvač, S. 1998. Formalizing Context (Expanded Notes). In *Computing Natural Language*, volume 81 of *CSLI Lecture Notes*. Center for the Study of Language and Information, Stanford University.

Mylopoulos, J., and Motschnig-Pitrik, R. 1995. Partitioning Information Bases with Contexts. In *3rd Int. Conference on Cooperative Information Systems.*

Subrahmanian, V. 1994. Amalgamating Knowledge Bases. *ACM Trans. Database Syst.* 19(2).