12-1991

Resource Integration Using a Large Knowledge Base in Carnot

Christine Collet

Michael N. Huhns
University of South Carolina - Columbia, huhns@sc.edu

Wei-Min Shen

Publication Info

Published in *IEEE Computer*, Volume 24, Issue 12, 1991, pages 55-62.
http://ieeexplore.ieee.org/servlet/opac?punumber=2

Resource Integration Using a Large Knowledge Base in Carnot

Christine Collet,* Michael N. Huhns, and Wei-Min Shen
Microelectronics and Computer Technology Corporation

This method for integrating separately developed information resources permits access and coherent modification. It uses the Cyc knowledge base as a global schema to resolve inconsistencies.

Today's corporate computing environments have many independent information resources. Because they must serve the needs of various applications, the resources might be of different types — for instance, a database management system with its databases, an information repository, an expert system with its knowledge base, or an application program with its data and productions.

These resources are largely incompatible in syntax and semantics, due not only to their different types, but also to diverse hardware and operating-system software, various physical and logical data structures, and contrasting corporate uses. Information resources attempt to model some portion of the real world, and in this attempt necessarily introduce simplifications and inaccuracies that lead to incompatibilities.

The goal of the research we describe in this article has been to develop a method for integrating separately developed information resources that overcomes these incompatibilities and permits the resources to be accessed and modified coherently. The method provides logical connectivity among the information resources via a semantic service layer that automates the maintenance of data integrity and provides an approximation of global data integration across systems. This layer is a fundamental part of the Carnot architecture,[1] which provides tools for interoperability across global enterprises.

The need for this capability is critical. Strategic business applications that require intercorporate linkage (for example, linking buyers with suppliers) or intracorporate integration (for example, producing composite information from engineering and manufacturing views of a product) are becoming increasingly prevalent. But creating such an environment requires that the incompatibilities, arising during query, update, and maintenance operations, be resolved.[2]

* Since collaborating on this article, Collet has accepted a professorship in France.

0018-9162/91/1200-0055$01.00 © 1991 IEEE

Two approaches to integrated access

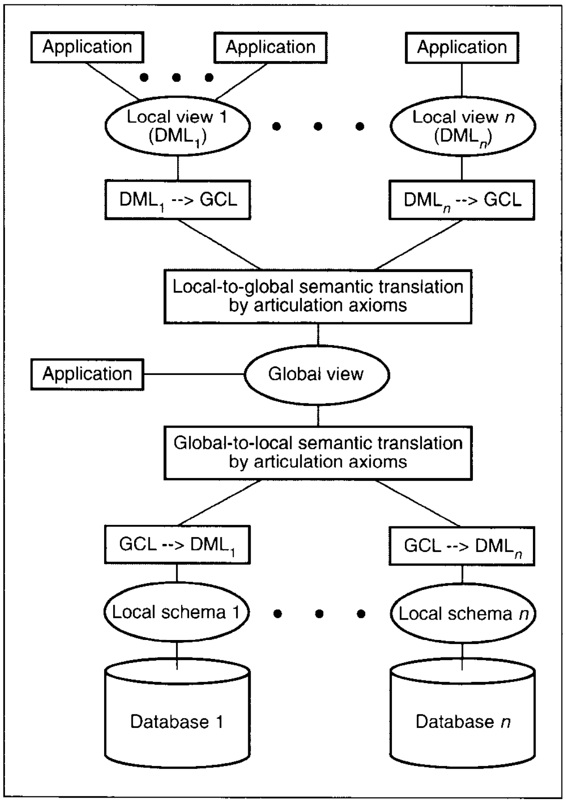


Figure 1. Global and local views in semantic transaction processing.

There are two general approaches for providing integrated access to a collection of heterogeneous databases.[3] They are called the *composite* approach and the *federated* or *multidatabase* approach.

The composite approach introduces a global schema to describe the information in the databases being composed. Database access and manipulation operations are expressed in a universal language and then mediated through the global schema.[4] Through this schema, users and applications are presented with the illusion of a single, centralized database. They need not be aware of semantic conflicts among the databases because explicit resolutions for the conflicts are specified in advance. However, the centralized view may be very different from the previous local views, so that existing applications might no longer execute correctly. Further, constructing a global schema is not only difficult but also must be repeated every time a local schema changes or is added.

The federated[5,6] or multidatabase[7] approach avoids constructing a global schema and merely presents a user with a collection of local schemas, along with tools for information sharing among the databases. The user resolves conflicts of facts in a manner particular to each application and integrates only the necessary portions of the databases. The advantages cited[5,7] for this approach include increased security, easier maintenance, and the ability to deal with inconsistent databases.

However, a user or application must understand the contents of each database to know what to include in a query; there is no global schema to provide advice about semantics. In addition, each database must maintain knowledge about the other databases with which it shares information. In Ahlsen and Johannesson,[8] this knowledge takes the form of models of the other databases, partial global schemas, a common data model, and an explicit agreement with each of the other databases. The number of local agreements and partial global schemas may be as high as $N(N-1)$, where N is the number of databases. By contrast, in the composite approach, only N mappings are required to translate between N databases and a global schema.

Our methodology for semantic integration is based on the composite approach, but our implementation differs in three ways, enabling us to combine the advantages of both approaches while avoiding some of their shortcomings.

First, rather than redo the global schema each time a new resource is to be integrated or a previously integrated resource is altered, we use an existing global schema — the Cyc knowledge base.[9] The schemas of individual resources are independently compared and merged with this knowledge base, although not with each other, making a global schema much easier to construct and maintain.

The Cyc knowledge base is the best available candidate for a global schema because of

- its large size (about 50,000 entities and relationships expressed as frames and slots), which covers a large portion of the real world and the subject matter of most information resources;
- its rich set of abstraction mechnisms, which ease the process of representing predefined groupings of concepts;
- its knowledge representation and inference mechanisms, which are needed to express the relationships among information resources and to construct, represent, and maintain a global schema; and
- its typing mechanism, which is used to integrate and check the consistency of query results.

Second, unlike most previous work on database schema integration, we use not only a structural description of the local schemas in resolving semantic differences but also all available knowledge, including

- *schema knowledge*, that is, the structure of the data, integrity constraints, and allowed operations;
- *resource knowledge*, that is, a description of supported services, such as the data model and languages, lexical definitions of object names, the data itself, comments from the resource designer, and guidance from the integrator; and
- *organization knowledge*, that is, the corporate rules governing use of the resource.

Third, the mapping between each individual information resource and the global schema is accomplished by a set of *articulation axioms*: statements of equivalence between components of two theories.[10] The axioms provide not only a semantic mapping between resources, but also a means of translation that enables the maintenance of a global view of all information resources and, at the same time, local views that correspond to each individual resource. An application can retain its current view but take advantage of some of the extra information that becomes available as information resources are integrated. Of course, any application can be modified to use the global view directly to access all available information.

The key aspects of our method are thus an ability to represent and integrate the semantics of individual information resources precisely, and an ability to maintain both global and local views. We describe an evaluation of our method based on the integration of three databases that have different data models (entity-relationship, relational, and object-oriented) but similar semantics for their data (that is, they capture information about the same domain). We also describe how the global and local views are used in semantic transaction processing.

Relations	Columns
AAAInfo	name* address rateCode lodgingType phone facility
AAADirection	address* direction
AAACredit	name* creditCard*
AAARate	name* season* 1P 2P1B 2P2B XP fCode

Figure 2. A relational database schema for the AAA tour book database.

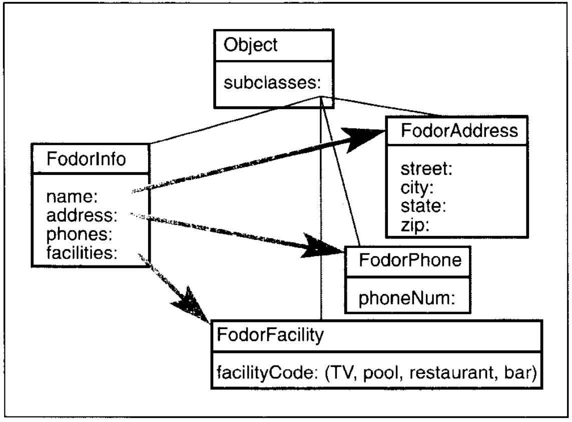


Figure 3. The object-oriented schema for the Fodor database.

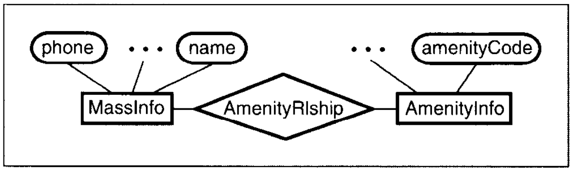


Figure 4. The entity-relationship schema for the Mass database.

Semantic transactions with global and local views

Resource integration is achieved by separate mappings between each information resource and the global schema (see Figure 1). Each mapping consists of a syntax translation and a semantics translation. The syntax translation provides a bidirectional translation between a local data manipulation language (DML_i) and the global context language (GCL), which is based on extended first-order logic. The semantics translation is a mapping between two expressions in GCL that have equivalent meanings. This is accomplished by a set of logical equivalences in GCL, called articulation axioms, having the form

$$ist(G\ \phi) \Leftrightarrow ist(S_i\ \psi)$$

where ϕ and ψ are logical expressions and *ist* is a predicate that means "is true in the context." This axiom says that the meaning of ϕ in the global schema G is equivalent to the meaning of ψ in the local schema S_i. At most, n sets of articulation axioms are needed to integrate n resources.

After integration, one can use the information that becomes available through a global view or a local view. The global view presents users and applications with the illusion of a single information resource, but they must use GCL, which might be unfamiliar.

The other option is the local view. Queries and updates can be issued against the local view, but they are not sent to any particular resource. Rather, they are first translated into the global language with terms that have global meanings, and then they are translated into different DML_i and distributed to appropriate information resources. The local view has the advantage that previous user knowledge and application programs do not need to be modified to access the extra but relevant information that becomes available. Note that external schemas previously defined on these local views remain intact, or new external schemas can be defined.

To illustrate the idea, we describe how transactions are processed semantically through the global and local views of three integrated databases.[11] The three databases have the same domain, that is, each contains information about hotels. However, they use different data models. The AAA database uses the relational model (see Figure 2). A hotel is called AAAInfo and represented as a relation. The features of hotel, such as name and address, are represented as columns.

The Fodor database uses an object-oriented data model (see Figure 3). A hotel is represented as a class called FodorInfo. The features of hotel are represented as fields of the class or as other object classes pointed at by FodorInfo.

The Mass database has an entity-relationship data model (see Figure 4). A hotel is represented as an entity called MassInfo, and its features are represented as attributes of this entity or as relationships to other entities. Note that these schemas represent different perspectives and different information about hotels.

Some of the Cyc concepts used in integrating these databases are the collections Lodging and Restaurant, and the predicates hasAmenities,* phoneNumber, and instanceOf. Example articulation axioms that map between the three database schemas and the global schema are

```
ist(G instanceOf(?H Lodging)) ⇔
   ist(AAA instanceOf
        (?H AAAInfo))
ist(G phoneNumber(?H ?P)) ⇔
   ist(AAA phone(?H ?P))
ist(G hasAmenities(?H ?F)) ⇔
   ist(AAA facility(?H ?F))
ist(G instanceOf(?H Lodging)) ⇔
   ist(Fodor instanceOf(?H
      FodorInfo))
ist(G hasAmenities(?H ?F)) ⇔
   ist(Fodor facility(?H ?X) ^
      facilityCode(?X ?F))
ist(G instanceOf(?A Restaurant)) ⇔
   ist(Mass instanceOf
        (?A AmenityInfo) ^
      amenityCode (?A 4))
```

Based on its local view of the AAA database, an application might issue the following query for the phone numbers of hotels that have a restaurant:

```
SELECT phone FROM AAAInfo
   WHERE facility = "Restaurant"
```

This local Structured Query Language query is first translated into GCL by the SQL-GCL syntax translator:

```
instanceOf(?L AAAInfo) ^
   instanceOf(?R Restaurant) ^
   facility(?L ?R) ^ phone(?L ?P)
```

This expression is then mapped by articulation axioms into a new expression whose semantics is meaningful in the global schema:

```
instanceOf(?L Lodging) ^
   instanceOf(?R Restaurant) ^
   hasAmenities(?L ?R) ^
   phoneNumber(?L ?P)
```

This is then translated into different local queries using the appropriate articulation axioms in reverse. The translation for the Fodor local schema is

```
instanceOf(?L FodorInfo) ^
   facilities(?L ?F) ^
   facilityCode(?F?R) ^
   instanceOf(?R Restaurant) ^
   phone(?L ?P) ^
   phoneNum(?P ?N)
```

The translation for the Mass local schema is

```
instanceOf(?L MassInfo) ^
   inAmenityRelationship(?L ?A)^
   instanceOf
   (?A AmenityRelationship) ^
   involvesAmenities
   (?A ?F) ^
   instanceOf
   (?F AmenityInfo) ^
   amenityCode(?F 4) ^
   phone(?L ?N)
```

These queries are then translated syntactically to appropriate local data manipulation languages before being sent to the databases. For example, the query sent to the Fodor database is the following object-oriented expression (using Itasca syntax):

```
(SELECT (FodorInfo phones
   phoneNum)
     (= (path (some facilities)
        (some facilityCode))
        "Restaurant"))
```

The query sent to the Mass database is an SQL-like entity-relationship expression:

```
SELECT phone FROM MassInfo,
   AmenityRelationship,
   AmenityInfo
     WHERE MassInfo.
        AmenityRelationship.
        amenityCode = 4
```

The SQL query sent to the AAA database is not generated from the global schema because it is the same as the original SQL query.

After the transactions are executed, the distributor assembles the results in the local view. In this example, the result is a column of phone numbers, because the local view of the AAA database is in the relational model. These phone numbers come from three databases and the list can be much longer than that from the AAA database alone. However, users and applications need not be aware of the extra sources of information.

The same process is used if a query is asked against the global view. In this case, the query is translated into three queries that are distributed to the three databases, and the results are in logic form (because the global view is described by first-order logic).

The global view is implemented as a combination of a transaction generator, a transaction distributor, and a result assembler. The transaction distributor is written in the actor language Rosette, which provides concurrency and asynchrony. The transactions are distributed to the resources, along with dependency properties among updated data and consistency requirements for these data (represented by eventual consistency, periodic consistency, or lagging consistency properties). We consider the execution of updates an important issue of schema integrity.

The development of articulation axioms

Articulation axioms for an information resource are developed in a three-phase process:

(1) schema representation,
(2) concept matching, and
(3) axiom construction.

The schema representation phase produces a Cyc context (microtheory[10]) containing a model for an information resource. In the second phase, concepts from the model are matched with appropriate concepts in Cyc's base context, the global schema (see Table 1). If there are no frames in the global schema corresponding to ones in the local schema, they are created. Matching is thus an interactive process. The user also might have to augment the model with additional properties (semantics) about the local schema for the matching phase to be completed. In the third phase, the matches are converted automatically into articulation axioms by instantiating templates for these axioms with terms from the matches.

Schema representation. The schema representation phase of the integration methodology represents the schema of

* In this article, the names of entities, objects, classes, relations, relationships, and Cyc collections are captialized, and the names of attributes, fields, and Cyc slots are not capitalized.

an information resource in the formalism of the global schema (see Table 1). The representation consists of a set of Cyc frames with slots residing in a separate context created for each information resource. These frames are instances of more general frames describing the data model used by the schema. For example, if we represent an instance of a relational schema, we will have frames of types Relation and DatabaseAttribute. The representation is structured using the properties described in the table in the "Semantics of resources" sidebar.

Table 1. Structures for representing database components.

Data Model Concept	Knowledge Base Structure
Object class	Collection
Object instance	Individual
Object attribute	Slot
Object method	Predicate
Entity-relationship entity	Collection
Entity-relationship relationship	Collection
Entity-relationship attribute	Slot
Relational table	Collection
Relational attribute	Slot
Relational tuple	Individual
Hierarchical segment	Collection
Hierarchical record	Individual
Hierarchical field	Slot
Codasyl record type	Collection
Codasyl record	Individual
Codasyl data item (Field)	Slot
Codasyl set (Link)	Slot

We define three types of frames for representing schemas:

- *DatabaseSchema frames*, describing the schemas for different data models;
- *DatabaseComponent frames*, describing the major components of schemas, such as relations and entities; and
- *DatabaseLink frames*, describing different kinds of links used to refine and relate the major components.

Every schema and every one of its components (relation,entity, attribute, etc.) is an instanceOf these types and belongs to a context characterizing that schema. For example, the schema Mass is represented as an instance of ERSchema in the context Mass, and every object of Mass is defined in this context. The slot dBSchemaMt, which is defined for the frame DatabaseSchema, is used to express the relationship between an instance of a schema and its context. Information about the use of a resource and the different functionalities (data definition language or DDL, DML, transactions) it provides are represented with the same approach as for schema representation, that is, using frames such as RelationalService, ERService, RelationalDDLType, and ERTransactionType.

Matching. The matching phase of integration can be considered the dual

Semantics of resources

The semantics of a resource can be obeyed and used more precisely if properties in addition to those contained in its schema can be specified. We encode in a knowledge base properties for specifying the semantics of both individual and collective resources.

The properties, shown in the accompanying table, provide a rich model for a resource and its components. A resource is viewed as a set of objects, along with the services to manipulate the set and the rules for their use within an organization. Objects can be concepts, models, data, integrity constraints, application programs, etc. The properties characterize concepts at the schema level and at the data or value level. They are instantiated during schema representation and are used during the subsequent matching phase of axiom development.

Little work in schema integration has been done on specifying the semantics of services and organizations to facilitate query decomposition, query optimization, and transaction management. Most work is based on the semantics of objects, primarily considering relationships among entities and attributes. Some of the relationships have been specified automatically by using heuristics[1] or applying subsumption.[2]

Properties for representing resource semantics.

Property	Applies To
Name	Schema object
Domain	Schema object
Format	Schema object
Makes-sense-for	Schema object
Documentation	Schema object
Integrity constraint	Schema object
Validation	Schema object
Synonym/homonym/ antonym	Schema object
Consistency	Schema object
Default value	Value object
Maximum value	Value object
Precision	Value object
Certainty	Value object
Name	Service object
Domain	Service object
Integrity constraints	Service object
Name	Organization object
Domain	Organization object
Availability	Organization object

References

1. J. de Souza, "SIS — A Schema Integration System," *Proc. Fifth British Nat'l Conf. Databases*, Cambridge University Press, 1986, pp. 167-185.

2. A.P. Sheth and S.K. Gala, "Attribute Relationships: An Impediment in Automating Schema Integration," *Proc. Nat'l Science Foundation Workshop Heterogeneous Databases*, 1989.

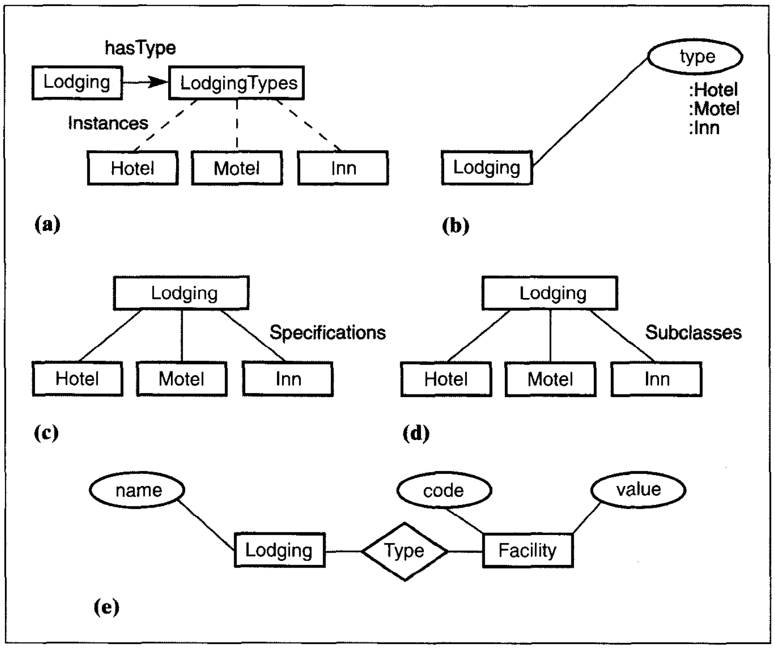


Figure 5. Possible representations in Cyc and resource schemas for the same concept, each allowing a different aspect of it to be emphasized: (a) Cyc concept represented as a slot; (b) Schema concept represented as an attribute; (c) Cyc concept represented as a category; (d) Schema concept represented as a class; and (e) Schema concept represented as a relationship.

problem to conceptual modeling for resource design (or to knowledge representation for expert system design). In conceptual modeling, the problem is, given a concept from the real world, find its model and representation. In resource integration, the problem is, given a (Cyc) representation for a concept, find its corresponding concept in the global schema. Several factors affect this phase. There might be a mismatch between the local and global schemas in the depth of knowledge representing a concept, and there might be mismatches between the structures used to encode the knowledge, as the example in Figure 5 shows. Specifically, matching is affected by

- *Concept representation in the local schema.* The local schema might use one of three different structures to represent a concept, corresponding to the primary structures of the relational, network or hierarchical, object-oriented, and entity-relationship data models.
- *Concept representation in the global schema.* The global schema might use different structures to represent a concept. In Cyc, a concept can be represented as either a category (a collection) or an attribute (see Lenat and Guha,[9] p. 339).
- *Relative knowledge.* The global schema might have more, less, or equivalent knowledge compared to a local schema. This factor applies to each concept in the local schema, rather than to the local schema as a whole.

If the global schema's knowledge is more than or equivalent to that of the local schema for some concept, the interactive matching process described in this section will find the relevant portion of the global schema's knowledge. This knowledge will be in one of Cyc's two forms for concept representation. If the global schema has less knowledge than the local schema, then knowledge will be added to the global schema until its knowledge equals or exceeds that in the local schema; otherwise, the global schema would be unable to model the semantics of the resource. The added knowledge refines the global schema.

Finding correspondences between concepts in the local and global schemas is a subgraph-matching problem. We base subgraph matching on simple string matching between the names or synonyms of frames representing the database schema and the names or synonyms of frames in the global schema. Matching begins by finding associations between attribute/link definitions and existing slots in the global schema. For example, matching the attribute definition numberOfRooms in the Mass context results in an association with the existing slot numberOfRooms in the Cyc global context.

After a few matches have been identified, either by exact string matches or by a user indicating the correct match out of a set of candidate matches, possible matches for the remaining schema concepts are greatly constrained (see the "Concept matching" sidebar). Conversely, after integrating an entity or an object, possible matches for its attributes are greatly constrained.

Unfortunately, string matching on names and synonyms is too weak a method for suggesting candidate matches. We are extending our matching mechanism to include other properties of the concept. For example, consider the integration of the attribute "other." The value of this attribute defines a description or a comment for an entity MassInfo, so its semantics are similar to the semantics of the Cyc slot "english."

The only means we see for finding such a similar slot is by

(1) accessing the value of the slot entryIsA of "other" to find its domain *C*,
(2) finding and listing all of the frames that have *C* as the value of their slot entryIsA, and
(3) asking the administrator to choose one from the list.

By considering additional properties for "other" given during the schema representation phase, such as its documentation, we can shrink the list of frames suggested for the concept.

Constructing articulation axioms. An articulation axiom is constructed for each match found. For example, the match that is found between the attribute numberOfRooms and the Cyc slot numberOfRooms results in the axiom

$$ist\ (G\ numberOfRooms(?L\ ?N)) \Leftrightarrow ist(Mass\ numberOfRooms(?L\ ?N))$$

meaning the numberOfRooms attribute definition determines the numberOfRooms slot in the global schema and vice versa. Articulation axioms are constructed automatically by using the matches to instantiate templates for the axioms, such as the templates shown in Table 2.

We have described an experiment in resource integration that we are conducting using the large knowledge base Cyc. Integration of resource schemas is based on articulation axioms defined between two contexts: a resource schema context and a global schema context provided by the Cyc knowledge base.

Our methodology is based on the following principles:

- Existing data should not have to migrate or be modified to achieve integration.
- Existing applications should not have to be modified due to integration.
- Users should not have to adopt a new language for communicating with the resultant integrated system, unless they are accessing new types of information.
- Resources should be able to be integrated independently, and the mappings that result should not have to change when additional resources are integrated.

These principles are incorporated in an integration tool for assisting an administrator in integrating a resource, and a transaction tool for providing users and applications with access to the integrated resources. The integration tool uses an extensive set of semantic properties to represent an information resource declaratively within the global schema and to construct bidirectional mappings between the resource and the global schema. The transaction tool uses the mappings to translate queries and updates written against any local schema into the appropriate form for each information resource. These tools constitute part of the semantic services of Carnot,[1] under development at the Microelectronics and Computer Technology Corporation. Carnot will enable development of open applications that can be tightly integrated with information stored on existing, closed systems. The semantic service layer of Carnot

Concept matching

Let a_{ij}, $j = 1, 2, ..., n$ denote the attributes of concept E_i in a local schema. E_i is the domain of the attributes, that is, the entity, relationship, relation, class, or object for which the a_{ij} are defined. Let s_j be the global schema slot that corresponds to, or matches, a_{ij}.

Observation 1: The domain C_j of slot s_j is a generalization of the concept in the global schema that matches E_i.

For example, the domain of the attributes numberOfRooms and phone is the entity MassInfo, whereas the domains of the corresponding Cyc slots numberOfRooms and phoneNumber are the frames HumanOccupyingStructure and Agent, respectively. These are generalizations of Cyc's Lodging, which is the frame whose semantics most closely corresponds to MassInfo.

As we match each of the attributes of E_i, we compute the common subdomain of the domains of their corresponding slots. The resulting common subdomains, although still generalizations of E_i, approximate it more and more closely.

Observation 2: The "best" match for E_i is $\cap_{j=1}^{n} C_j$, the most general common subdomain (greatest lower bound in the generalization hierarchy) of the slot domains.

In the example above, the most general common subdomain of HumanOccupyingStructure and Agent is ServiceOrganization, a generalization of Lodging. This would be suggested as the approximate match for MassInfo. If no other attributes are matched, this would also be the best match that could be determined automatically for MassInfo.

The greatest lower bound might not exist as a single frame in the global schema, however; it might be a set of frames. For example, the greatest lower bound would be the set {HumanOccupyingStructure Agent} if the frame ServiceOrganization did not exist. In such a case, a frame would be created in the Cyc knowledge base with the frames in the set listed as its generalizations.

Table 2. Templates for building articulation axioms.

Cyc Concept Represented	Schema Concept Represented	Articulation Axiom Template
As slot	As attribute, *a*	$ist(G\ iO(x\ C_1) \Leftrightarrow ist(LC\ iO(x\ E_1)$ $\wedge s(x\ y)) \qquad \wedge a(x\ y))$
As slot	As class	$ist(G\ iO(x\ C_1) \Leftrightarrow ist(LC\ iO(x\ y))$ $\wedge s(x\ y))$
As slot	As relationship, *R*	$ist(G\ iO(x\ \mathrm{C}_1) \Leftrightarrow ist(LC\ iO(x\ E_1)$ $\wedge iO(y\ C_2) \qquad \wedge iO(y\ E_2)$ $\wedge s(x\ y)) \qquad \wedge iO(z\ R)$ $\wedge r_1(x\ z)$ $\wedge r_2(z\ y))$
As category	As attribute	$ist(G\ iO(x\ C_2) \Leftrightarrow ist(LC\ iO(x\ E_1)$ $\wedge specs(C_1\ C_2)) \qquad \wedge a(x\ E_2))$
As category	As class	$ist(G\ iO(x\ C_2) \Leftrightarrow \quad ist(LC\ iO(x\ E_2)$ $\wedge specs(C_1\ C_2)) \quad \wedge subclasses(E_1\ E_2))$
As category	As relationship	$ist(G\ iO(x\ C_2) \Leftrightarrow ist(LC\ iO(x\ E_2)$ $\wedge specs(C_1\ C_2)) \qquad \wedge iO(y\ E_1)$ $\wedge iO(z\ R)$ $\wedge r_1(y\ z)$ $\wedge r_2(z\ x))$

Notes: These axioms assume that global entity C_i matches local entity E_i. *iO* denotes instanceOf, *LC* denotes the local schema context, and *specs* denotes Cyc's subclass relation.

provides facilities to specify and maintain the semantics of an organization's integrated information resources.

A major problem we have not yet solved is how to integrate the results returned from multiresource queries. This integration requires both the types and the formats of the results. Unfortunately, attribute values are not usually typed objects in databases, and only their format is typically specified in schemas. The knowledge representation language used for the global schema is strongly typed, however, and provides a basis for extending the articulation axioms to be used for mapping and integrating results. We are now making this extension.

Another problem involves the size of the global schema. Because it is significantly larger than the strict set of frames involved in integrating the databases, a user can issue more-general queries than if a merged schema were used. However, its large size also makes it difficult to use.

Through this article, we have shown that a user or an application can continue to use a familiar local schema and still benefit from resource integration. In addition, we are developing a graphical entity-relationship representation of the global schema and an intelligent interface for specifying queries with global semantics. ■

Acknowledgment

Christine Collet worked on this project at the Microelectronics and Computer Technology Corporation under a fellowship sponsored by Institut National de Recherche en Informatique et en Automatique, Paris.

References

1. P.E. Cannata, "The Irresistible Move Towards Interoperable Database Systems," *Proc. First Int'l Workshop Interoperability Multidatabase Systems*, IEEE CS Press, Los Alamitos, Calif., Order No. 2205, 1991, pp. 2-5.

2. A.P. Sheth and J.A. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," *ACM Computing Surveys*, Vol. 22, No. 3, Sept. 1990, pp. 183-236.

3. O.P. Buneman, S.B. Davidson, and A. Watters, "Querying Independent Databases," *Information Sciences*, Vol. 52, Dec. 1990, pp. 1-34.

4. T. Landers and R.L. Rosenberg, "An Overview of Multibase," *Distributed Data Bases*, H.J. Schneider, ed., North-Holland, Amsterdam, 1982, pp. 556-579.

5. D. Heimbigner and D. McLeod, "A Federated Architecture for Information Management," *ACM Trans. Office Information Systems*, Vol. 3, No. 3, July 1985, pp. 253-278.

6. S.B. Navathe et al., "Federated Architecture for Heterogeneous Information Systems," *Proc. Nat'l Science Foundation Workshop on Heterogeneous Databases*, 1989.

7. W. Litwin, L. Mark, and N. Roussopoulos, "Interoperability of Multiple Autonomous Databases," *ACM Computing Surveys*, Vol. 22, No. 3, Sept. 1990, pp. 267-293.

8. M. Ahlsen and P. Johannesson, "Contracts in Database Federations," in *Cooperating Knowledge Based Systems 1990*, S.M. Deen, ed., Springer-Verlag, London, 1991, pp. 293-310.

9. D. Lenat and R.V. Guha, *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*, Addison-Wesley, Reading, Mass., 1990.

10. R.V. Guha, "Microtheories and Contexts in Cyc Part I: Basic Issues," MCC Tech. Report No. ACT-CYC-129-90, June 1990.

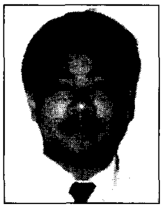
11. R. Wang and S. E. Madnick, "Facilitating Connectivity in Composite Information Systems," *Data Base*, Fall 1989, pp. 38-46.

Christine Collet formerly served as a visiting scientist at the Microelectronics and Computer Technology Corporation, Austin, Texas, working under an INRIA fellowship. She is now an associate professor at the University of Joseph Fourier, Grenoble, France. Her research interests are in databases, especially data models, schema integration, database programming languages, fourth-generation languages, and integrity constraint management.

Collet completed her PhD in 1987 on the management of dynamic complex objects (forms) at the University of Grenoble. The form model offers concepts for specifying the structural, semantic, and presentation aspects of complex objects. This model has been implemented on top of the object-oriented database system O2.

Michael N. Huhns is a senior staff member of the Artificial Intelligence Laboratory at the Microelectronics and Computer Technology Corporation, Austin, Texas, where he has been conducting research on the Argo, Antares, RAD, and Carnot projects. He has been with MCC since 1985. His research interests are in machine learning, distributed artificial intelligence, database schema integration, and computer vision.

Huhns received the BSEE degree from the University of Michigan in 1969, and the MS and PhD degrees in electrical engineering from the University of Southern California in 1971 and 1975, respectively. He has written 87 technical papers in machine intelligence and edited volumes I and II of *Distributed Artificial Intelligence*. He is a member of the IEEE, the IEEE Computer Society, Sigma Xi, Tau Beta Pi, Eta Kappa Nu, ACM, and AAAI.



Wei-Min Shen is a staff member of the Artificial Intelligence Laboratory at the Microelectronics and Computer Technology Corporation, Austin, Texas, where he has been conducting research on the Cyc and the Carnot projects. He has been with MCC since 1989. His research interests are machine learning and discovery, autonomous agents, database schema integration, and robotics.

Shen received the BS degree in computer science and engineering from the Northern JiaoTong University, Beijing, China, in 1982, and the MS and PhD degrees in computer science from Carnegie Mellon University in 1986 and 1989, respectively. He is the author of 18 technical papers in artificial intelligence and the forthcoming book *Autonomous Learning from the Environment*. Shen is a member of the ACM and AAAI.

Readers can contact Collet at LGI/IMAG, BP 53X, 38041 Grenoble, France; and Huhns and Shen at the Microelectronics and Computer Technology Corp., 3500 W. Balcones Center Dr., Austin, TX 78759-6509.