WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

INTEGRATING DISPARATE DATABASES FOR COMPOSITE-ANSWERS

Stuart E. Madnick

Y. Richard Wang

September 1987 #WP 1936-87

MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139

INTEGRATING DISPARATE DATABASES FOR COMPOSITE-ANSWERS

Stuart E. Madnick

Y. Richard Wang

September 1987 #WP 1936-87

To appear in Proceedings of the Hawaii International Conference on System Sciences, January 1988.

Integrating Disparate Databases For Composite-Answers

Stuart E. Madnick
Y. Richard Wang*
E53-320, Sloan School of Management
MIT, Cambridge, MA 02139
(617) 253-6671

* On leave from the MIS Department, College of BPA,
University of Arizona, Tucson, AZ 85721

ABSTRACT Many important information systems require multiple independent databases to work together within and/or across organizational boundaries in order to increase productivity. We refer to this type of systems as Composite Information Systems (CIS). A key area of research in CIS is *logical connectivity* which deals with the process of accessing disparate databases in concert for composite-answers. The major problems that need to be overcome to attain logical connectivity include contradiction, inconsistency, and ambiguity.

This paper presents an approach to resolve these problems through enhancing the semantic power of the database integration. This approach exploits concepts drawn from frame-based knowledge representation and rule-based inferencing. An object-oriented prototype is also presented to illustrate the process involved in formulating composite-answers where different levels of abstraction are required.

KEY WORDS AND PHRASES: Distributed Database Systems, Object-Oriented Programming, Organizational Information Systems, Strategic Computing.

ACKNOWLEDGEMENTS Work reported herein has been supported in part by the Department of Transportation's Transportation System Center, the U.S. Air Force, the Center for Management of Information at the University of Arizona, and Citibank.

1. Introduction

Significant advances in the price, speed-performance, capacity, and capabilities of new database technology have created a wide range of opportunities for commercial applications. These opportunities are widely recognized now as strategically important to corporations. It is also increasingly evident that the identification of strategic applications alone does not result in success for an organization. A careful and delicate interplay between choice of strategic applications, appropriate technology, and organizational responses must be made to attain success, as depicted in Figure 1 [2]. An effective information system is one that successfully align the problems and opportunities across these three domains.

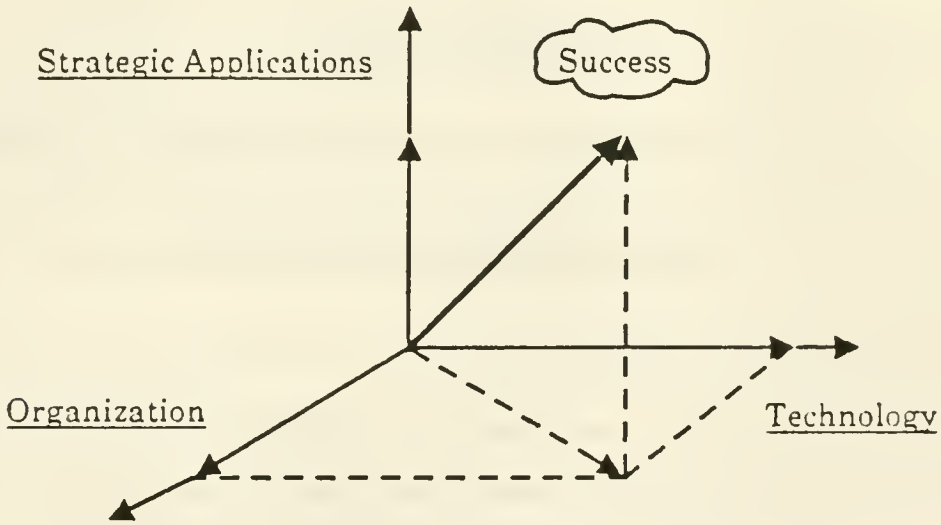

Figure 1 A Strategic Applications, Technology, and Organizational Research Initiative (SATORI)

One important category of strategic applications involve inter-corporate linkage (e.g., tying into supplier and/or buyer systems) and/or intra-corporate integration (e.g., tying together disparate functional areas within a firm). This category of systems is referred to as *Composite Information Systems* (CIS) [2, 6, 14] hereinafter.

This paper addresses issues involved in the evolution of separate systems to a more fully integrated CIS in order to gain strategic advantage. Benefits of CIS and recent research activities are presented in the remainder of this section. Section 2 provides a case study of multiple tour-guide databases to illustrate the strategy for semantic integration of disparate database systems. Section 3 presents a solution that combines Data Base Management System (DBMS) and Artificial Intelligence / Expert Systems (ALES) techniques to facilitate logical connectivity. In particular, concepts drawn from frame-based knowledge representation and rule-based inferencing are exploited. Furthermore, an object-oriented prototype system was developed to test this approach. Finally, concluding remarks are made in section 4.

Benefits of CIS

A key idea of CIS is the integration of disparate databases for composite answers. Without integration, it is difficult, expensive, time-consuming, and error-prone to obtain key information which may be distributed in databases located in different divisions of different organizations.

Consider the following case study of a major international bank [2], as shown in Figure 2. Currently, three separate database systems are being used for loan management, cash management, and letters-of-credit processing. Suppose a client requests that $50,000 be transferred to another account. If that client's cash balances in the funds transfer system can not cover that transaction, it will be rejected -- even though that client may have a $20,000,000 active letter-of-credit! This rejection, besides being annoying to the client, will require significant effort to correct by manually drawing on the letter-of-credit to cover the funds transfer. If the bank can connect the three separate database systems together to access information in concert, so that funds can be automatically drawn on the letter-of-credit, then

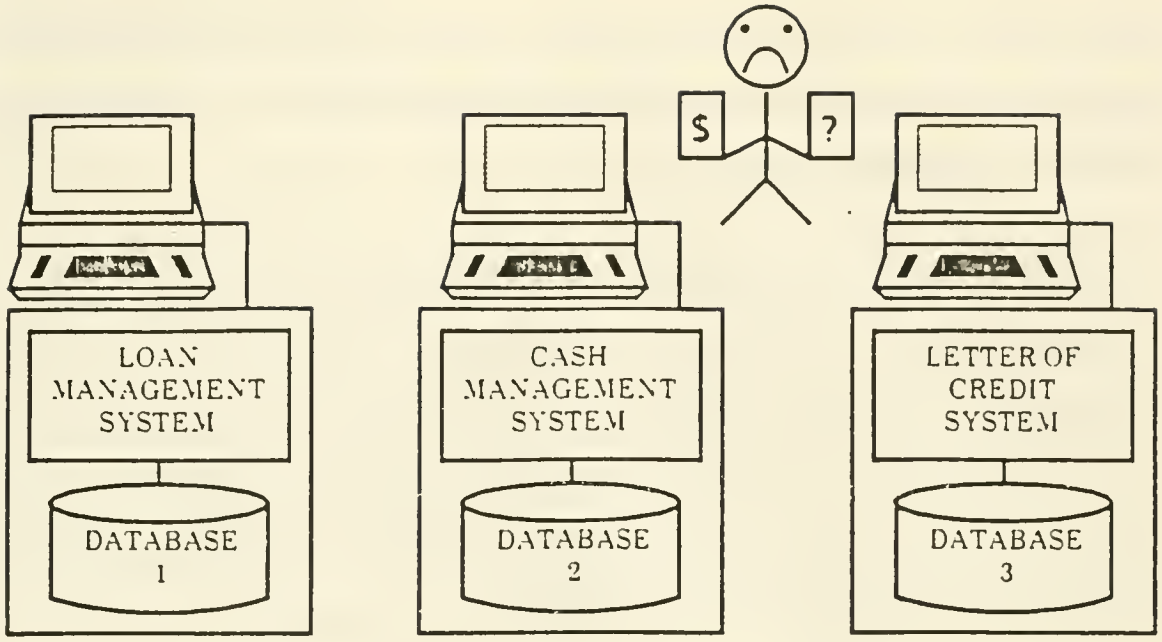

Figure 2 An Electronic Banking System Without Integration

product-differentiation will be achieved via the enhanced quality of service, and reprocessing costs can be reduced since special manual intervention can be avoided.

Recent Research Activities

Researchers in the DBMS and AI/ES fields have been striving to enhance their systems in a converging direction. The database community seeks additional semantic capability in their DBMS to "understand" more of the real world, while the AI/ES community has begun to address the increasing demand for database access. This converging trend has prompted researchers to address the connectivity problem from different perspectives. Furthermore, twelve methodologies for database schema integration have been analyzed and compared by Batini, Lenzerini, and Navathe [1]. We summarize some recent research activities that benefited our work below.

The *open-systems* group [5] deals with highly parallel, distributed, open systems. It is based on the belief that future computer applications will involve the

interaction of subsystems that have been independently developed at disparate geographical locations. Message Passing Semantics is a methodology being developed to tackle connectivity.

The *MULTIBASE research project* [3, 15] attempts to provide a uniform interface through a single query language and database schema to data in pre-existing, heterogeneous, distributed databases. The Functional Data Model and the data language DAPLEX are used as the common data model and language to efficiently execute queries that may require data from different databases with different schemata, data models, and query languages.

The *federated architecture research* [4, 9] aims at uniting a collection of independent database systems into a loosely-coupled federation. The export-schema specifies the information that a component will share with other components, while the import-schema specifies the non-local information that a component wishes to manipulate.

Each of these research efforts has addressed certain aspects of connectivity. In comparison, the research goals of the CIS research project [2, 6, 14] is twofold: (a) to develop a methodology for integrating disparate database systems with a particular focus on *logical connectivity*; and (b) to demonstrate the feasibility of a system capable of attaining logical connectivity in a real environment with satisfactory performance.

2 Developing CIS

Recent advances in DBMS and AI/ES have provided a new perspective to the design of CIS. At the external level, query languages such as QUEL and Query-By-Forms (QBF) in INGRES have been employed to provide user-friendly interfaces. Although these interfaces are powerful, the user still needs to follow certain formats. Since the purpose of accessing multiple databases is to formulate composite answers in order to achieve the user's objective, it wil be even more powerful if the user can simply "query by objective (QBO)." A CIS should allow the user to query by a statement-of-objective which is processed by an *Intelligent Front-End Processor* (IFEP). IFEP eventually produces a composite-query executable by the underlying system.

At the conceptual level, a composite-query will be processed to produce the required information which may be distributed in disparate databases located in different divisions of different organizations. Integrating these disparate databases requires knowing where all the data are stored, facilitating all the necessary interfaces, constructing queries to retrieve the data, accumulating local results, resolving local differences, and integrating the results into composite-answers.

MULTIBASE [15], for example, used a Global Data Manager (GDM) and Local Database Interfaces (LDI) to provide a uniform interface. Many other methodologies for database integration have been proposed in the past decade. They are categorized into two contexts [1]: (a) *view integration* which produces a global conceptual description of a proposed database; and (b) *database integration* which

(1) Multi-database interoperability has also been proposed as an alternative approach. It assumes that the databases the user may access basically have no global schema. The system should provide the user with functions for manipulating data that may be in visibly distinct schemata and may be mutually nonintegrated [8].

produces the global schema of a collection of databases. This global schema is an integrated view of all databases in a distributed database environment [1].

To develop strategic applications of databases *quickly* in order to obtain tangible benefits before a full-scale heterogeneous database integration is launched, the concept of *virtual-drivers*[2] can be applied, as delineated below. Note that the virtual-driver concept is equally applicable in an environment where each component is a distributed database management system such as R* or INGRES*.

Virtual-Drivers

In the virtual-driver concept [14], users can still use real terminals to perform their traditional functions, as shown in Figure 3. Meanwhile, virtual-drivers are created which are indistinguishable to the system from users of real terminals. A user interested in obtaining composite-answers from multiple databases invokes the CIS-Executive which mediates the virtual-drivers. The CIS-Executive invokes each system (via its virtual-driver) to obtain the necessary information. Incompatibilities among the database systems are resolved by the CIS-Executive which then presents a composite-answer to the user. Two levels of connectivity need to be considered when using the virtual-driver approach, as discussed below.

Physical connectivity refers to the process of actual communication among disparate databases. Although many R&D issues need to be addressed in physical connectivity (e.g., bandwidths, security, availability, and reliability) in order to insure an effective CIS, we assume that communication solutions are available, and focus on *Logical connectivity* which is even more challenging. Logical connectivity

(2) It is important not to confuse the concept of virtual-driver with virtual-terminal protocols. The virtual-terminal protocols have been invented to try to hide terminal idiosyncracies from application programs through mapping of real terminals onto a hypothetical network virtual terminal. In contrast to the narrow mapping, the virtual-driver concept aims at accessing separate databases in concert to formulate composite answers.

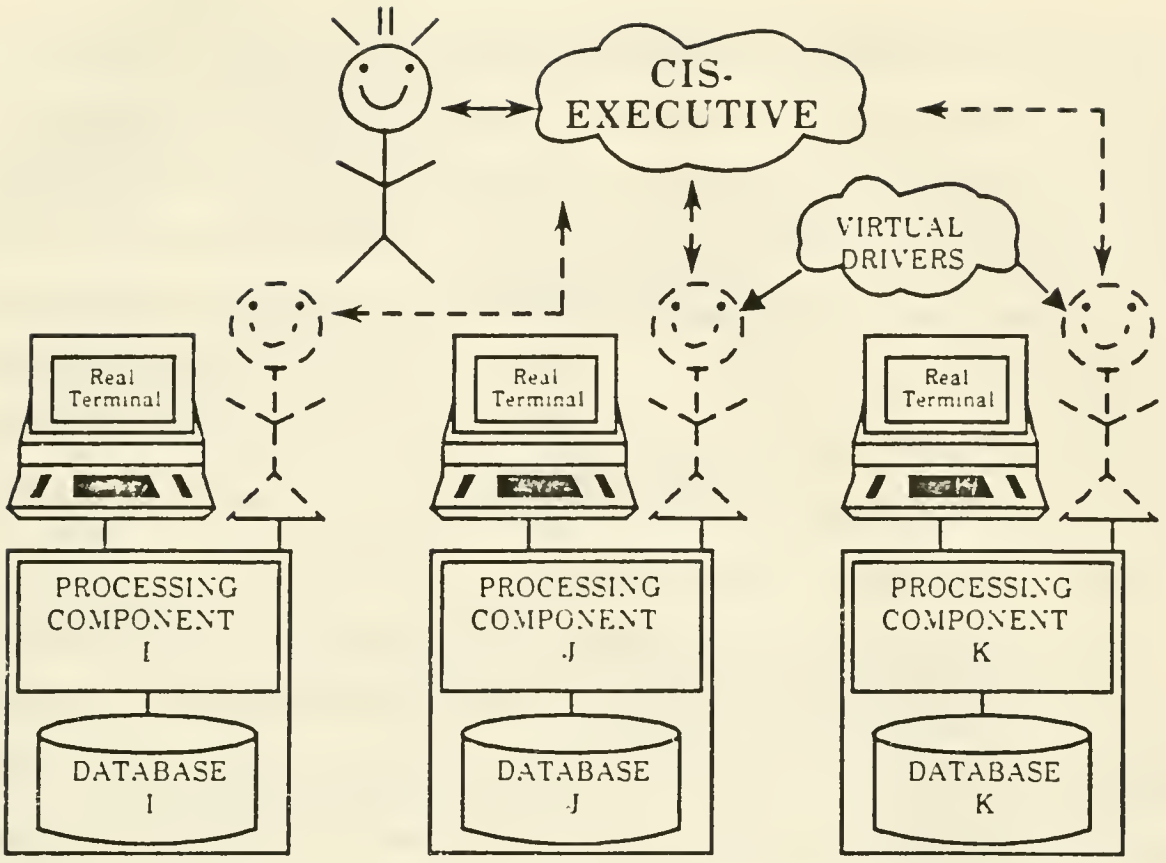

Figure 3 Integrating Disparate Databases

refers to the process of accessing disparate databases in concert for composite-answers. The major problems that need to be overcome to attain logical connectivity include syntax and semantic contradiction, inconsistency, and ambiguity due to different assumptions made in disparate databases. For brevity, *connectivity is used instead of logical connectivity* hereinafter. A travel-guides case is used to illustrate connectivity strategy.

Connectivity Strategy

Travel-guides are easy to understand, abundant in data-semantics, and representative of the situation involved in CIS. Litwin and Abdellatif reported a prototype multi-database system using tour-guides for Paris as an example [8]. The conventional DDL/DML is extended in their prototype to incorporate the

complexities introduced by the explicit recognition that a global schema is not available. Similar to MULTIBASE, connectivity is attempted in Litwin and Abdellatif's prototype via DDL/DML. In other words, connectivity is encoded by the DBA or application programmers; therefore, opaque to the end-user. In contrast, we exploit concepts drawn from frame-based knowledge representation scheme and rule-based inferencing as a strategy to attain connectivity. An advantage of this approach is that questions such as "why and how" can be answered upon the request of the end-user, making the connectivity-process more transparent to the end-user. Other advantages will be delineated along the way.

Three tour-guides are presented below to illustrate connectivity strategy. They are AAA Tour-book for Massachusetts. 1987 (abbr. AAA hereinafter), the FODOR's New England. 1987 (abbr. FODOR), and The Spirit of Massachusetts, 1987 (abbr. MASS). One might envisage that AAA is implemented in INGRES* where each state has a local physical database transparent across the states. Similarly, FODOR is implemented in distributed ORACLE, and MASS in R* by competing organizations. *Our focus is on the semantics of each database*, not the DBMS used to implement the database.

Interacting with a workstation which uses an CIS-Executive to drive these distributed databases, a travel agent may seek to meet th tement-of-objective established by a client in her office. Suppose that the client s statement-of-objective is to maximize the quality of facility with minimum cost. Suppose also that part of the composite-query produced by IFEP is to obtain a composite-answer of the facilities of Logan Airport Hilton in Boston from the tour-guide databases. Let us see how we may formulate a composite-answer for the query "what are the facilities of Logan Airport Hilton in Boston" from the tour-guide databases with database schemata shown in Figure 4.

AAA Relations

AAA-Info:	(Name*, Address, Rate-Code, Lodging-Type, Classification, #-of-Units, Phone#, Other)
AAA-Direction:	(Address*, Direction)
AAA-Facility:	(Name*, Facility*)
AAA-Credit:	(Name*, Credit-Card*)
AAA-Rate:	(Name*, Season*, 1PL, 1PH, 2P1BL, 2P1BH, 2P2BL, 2P2BH, XP, F-code)

FODOR Relations

FODOR-Info:	(ID#*, Name, Address, Comment, Location, Package,Category)
FODOR-Phone:	(ID#*, Phone#*)
FODOR-Facility:	(ID#*, Facility*)
FODOR-Service:	(ID#*, Service*)

MASS Relations

MASS-Info:	(Name*, Address, Facility-Tytpe, Rating, #-of-Rooms, Other)
MASS-Phone:	(Name*, Phone#*)
MASS-CC:	(Name*, CC*)
MASS-Amenity:	(Name*, Amenity-code*)
MASS-Package:	(Name*, Package-Name*, Package-Descript)

Figure 4 Relational Schemata for AAA, FODOR, and MASS

In order to retrieve the data and the data-formats of the facilities, the schemata and data dictionaries need to be accessed. The COLUMNS in the data dictionary of MASS are exemplified in Table 1. The CIS-Executive can be designed to decompose the query (i.e., "What are the facilities of Logan Airport Hilton in Boston?") into subqueries which perform operations such as SELECT, JOIN, and PROJECT on the relations in AAA, FODOR, and MASS. As a result, data of Logan Airport Hilton can be accumulated, as shown in Table 2. The reader may have recognized that it is necessary to realize that *amenity* in MASS equates to *facility* in FODOR and AAA. In addition, the amenity-code in MASS has to be converted (e.g., 6 means pool) in order to construct the MASS-column in Table 2. *Although such mappings are not automatic on most current DBMS, these transformations are fairly straightforward* and are performed by experimental systems such as MULTIBASE.

Pursuing the case further, we observe that FODOR reports less information than AAA and MASS. It is possible that FODOR simply does not emphasize the facilities as the other two guides, but specializes in other aspects such as decor,

TNAME	CNAME	COLTYPE /LENGTH
MASS-Info	Name	Char(30)
MASS-Info	Address	Char(50)
MASS-Info	Facility-Type	Num(1)
MASS-Info	Rating	Char(4)
MASS-Info	#-of-Rooms	Num(2)
MASS-Info	Other	Char(80)
MASS-Phone	Name	Char(30)
MASS-Phone	Phone#	Char(13)
MASS-CC	Name	Char(30)
MASS-CC	CC	Char(2)
MASS-Amenity	Name	Char(30)
MASS-Amenity	Amenity-Code	Num(1)
MASS-Package	Name	Char(30)
MASS-Package	Package-Name	Char(40)
MASS-Package	Package-Descript	Char(80)

Table 1 COLUMNS in the Data Dictionary of MASS

location, and historical anecdotes. It turns out that from FODOR-Info in Figure 4, it would be easy to know that Logan Airport Hilton is categorized as "expensive" (via a query to SELECT from FODOR-Info where name = "Logan Airport Hilton," followed by a PROJECT on category). Moreover, FODOR interprets "expensive" as quoted below.

"All rooms must have bath or shower, valet and laundry service, restaurant and bar, at least some room service, TV and telephone in room, attractive furnishings, heating and air conditioning."

Therefore, new information can be *inferred* and added to Table 2, making Table 3. It is worth noting that "category" is used in FODOR (reflected in Figure 4) instead of "classification" in AAA or "rating" in MASS. Note also that the meaning of "category" is *not stored* as part of the relations. Consequently, it is necessary to embed the knowledge somewhere, and automatically invoke the appropriate

AAA	FODOR	MASS
Parking lot		Free parking
C/TV		Cable TV
A/C		Air Conditioning
Phones		Telephone in room
Pool	Outdoor pool	Pool
Airport transport	Airport car avail.	Free transportation to/from airport
Dining rm	Restaurants	Restaurant
Non-smokers' room		Non-smoker rooms
	Pets	Pets allowed
Cocktails	Bar	Lounge
Suites	Entertainment	Near public transportation
Fee for movies	Dancing wknds	Handicapped accessible
350 radios		
Smoke detectors		

Table 2 Data for Logan Airport Hilton (Without Rating)

mechanism to retrieve the knowledge when needed. Furthermore, the data accumulated in Table 3 reflect different assumptions embedded in the databases being integrated due to different mental models used by different designers. Resolving these semantic issues represents the difficult challenge for comprehensive connectivity.

We summarize some of the issues raised.

- **Synonym.** Type-of-lodging such as hotel, motel, and inn in AAA are referred to in MASS as type-of-facilities. "Category" in FODOR is referred to as "classification" in AAA, and "rating" in MASS.
- **Conversion.** In MASS, the amenity-code 6 is pool, and 9 is cable-TV.
- **Format.** In AAA, the data format of facility is in characters, whereas numeric-code is used in MASS, as shown in Table 3.
- **Incompleteness.** Each travel-guide provides partial information regarding facility at Logan Airport Hilton in Boston. Furthermore, each guide specializes

AAA	FODOR	MASS
Character 25 +	Character 30 +	Numeric 1 +
Parking lot		Free parking
C/TV	TV in room*	Cable TV
A/C	A/C*	Air Conditioning
Phones	Phone in room*	Telephone in room
Pool	Outdoor pool	Pool
Airport transport	Airport car avail	Free transportation to/from airport
Dining rm	Restaurants	Restaurant
Non-smokers' room		Non-smoker rooms
	Pets	Pets allowed
Cocktails	Bar	Lounge
Suites	Entertainment	Near public transportation
Fee for movies	Dancing wknds	Handicapped accessible
350 radios	Heating*	
Smoke detectors	Bath or shower*	

Table 3 Data for Logan Airport Hilton With Rating

\+ indicates the data formats of the attributes
* indicates that the facility is inferred from the FODOR rating

in certain aspects of the problem domain. For example, AAA specializes in room-rate, FODOR specializes in decor and location, and MASS specializes in handicapped service.

- **Granularity.** FODOR simply reports whether TV is available or not, but AAA has three categories for TV (i.e., C/TV for color TV; CATV for cable TV; and C/CATV for color cable TV). The level of granularity may lead to semantic differences as illustrated below.
- **Contradiction.** AAA indicates that Logan Airport Hilton has color TV without cable, whereas MASS reports that cable TV is available (In reality, it has color TV with paid movies for special stations such as HBO).
- **Ambiguity.** It is certainly not clear whether the TV-availability in FODOR means color (and/or cable) or not. Similarly, room-rate in different travel-

guides has very different meanings. Example sources of difference include whether tax, breakfast, service charge, and granuities are included or not.

- **Inconsistency.** The name, address, and phone number are reported as follows:

  AAA: Logan Airport Hilton; Logan International Airport, East Boston, 02128 (617) 569-9300

  FODOR: Hilton Inn at Logan: Logan Int'l Airport, 569-9300

  MASS: The Logan Airport Hilton; Logan International Airport, Boston, 02128 (617) 569-9300

The twelve methodologies reported by Batini et al. [1] also dealt with some subset of these issues. This tour-guides case presented above provides a cohesive setting to discuss these issues and the corresponding connectivity strategy, as elaborated below.

To resolve these issues, it is necessary to map synonyms and convert different data formats. This can be accomplished, in part, through data dictionaries. View definition and integration techniques can be applied to provide a more comprehensive view from the partial, incomplete information in the local databases. Through comparison, conforming, and integration of the schemata, a global conceptual schema may be proposed and tested against the following criteria: (a) completeness and correctness; (b) minimality; and (c) understandability [1].

This process requires (a) extensive manual effort, and (b) the resulting integrated schema does not contain explicit knowledge of the assumptions made. To resolve these two problems, it is necessary to understand the concepts that underlie the data. With these concepts in mind, reasonable connections may be established based on the content of the database(s). We refer to this type of problem as *semantic reconciliation*. Connectivity strategy can be employed to solve semantic-reconciliation problems in order to formulate composite-answers. Two parts of this

strategy are illustrated below: (a) identifying the same object without a global identifier; and (b) making a judgment call based on credibility.

(A) Identifying the same object without a global identifier

A *unique global key identifier may not exist when multiple databases are involved.* For example, there does not exist a primary key identifier which is defined consistently and completely across the tour-guide databases. In order for the subqueries (decomposed by the CIS-Executive discussed earlier) to perform the necessary database operations, a unique ID for "Logan Airport Hilton" is needed, be it an ID-number, a name, an address, or a phone-number. Note that the name used in FODOR is "Hilton Inn at Logan." If "Logan Airport Hilton" is used as the identifier to retrieve facility from the FODOR database, nothing will be returned.

A moment of thought would lead one to make the connection between phone numbers and a hotel. Let us assume that a hotel may have more than one phone, but does not share the same phone number with another hotel. Using "Logan Airport Hilton" as the name for the hotel, the phone number (617) 569-9300 could be retrieved from AAA given the relations in Figure 4. Using (617) 569-9300 as a reference, the phone numbers 569-9300 for FODOR, and (617) 569-9300 for MASS could also be retrieved.

The remaining question is whether "617" is also the area code for 569-9300 in FODOR. From FODOR, the hotel which has the phone number 569-9300 is classified as located in the Boston area, which has an area-code 617. Consequently, the three hotels have exactly the same phone number, i.e., (617) 569-9300. Since a hotel does not share the same phone with another by assumption, we conclude that it is the same hotel. As a result, facility data can be retrieved from the databases.

Thus, to identify the same object in disparate databases without an explicit global identifier requires us to make extensive use of inference with knowledge in the application domain.

(B) Making a judgment call based on credibility

Contradiction, granularity, inconsistency, and ambiguity are unavoidable when integrating disparate databases. What should we do when different sources of information disagree? How can we make our system sensitive to different perspectives? An informed approach based on the *credibility* of the local databases can help the delivery process. The fact that AAA specializes in room-rate, FODOR specializes in decor and location, and MASS specializes in handicapped service is an example of *credibility* which can be used to make a judgment call when needed. We give another concrete example below.

As mentioned earlier, AAA indicates that Logan Airport Hilton has color TV without cable but MASS reports that cable TV is available, contradicting each other. A closer examination reveals that AAA is more specific. It has three categories for TV: C/TV for color TV, CATV for cable TV, and C/CATV for color cable TV. Whereas MASS only indicates if cable TV is available. Therefore, it is fair to say that AAA is *more detailed*; therefore, more likely to be *more credible* in reporting TV information. In addition, AAA offers fee for movies (assuming that the CIS-Executive is able to interpret the data accumulated in Table 3). If the CIS-Executive also "knows" that cable TV means fee, then it would conclude that "Cable TV" in MASS means "fee for cable movies such as HBO." That is, it has color TV without cable, but offers fee for movies from special stations such as HBO.

Based on the analysis presented in this section, a composite-answer is formulated below for the facilities of Logan Airport Hilton. The reader may wish to propose variations of the composite-answer.

"(1) free parking; (2) color TV without cable, but with fee for movies, e.g., HBO; (3) air conditioning; (4) phone in room; (5) pool; (6) airport transportation available; (7) restaurant; (8) non-smokers' room; and (9) pets allowed. In addition, the following facilities have been reported suites, smoke detectors, entertainments, dancing weekends, bath or shower, attractive furnishing, cocktail bar lounge, near public transportation, and handicapped accessible."

The tour-guide case has shed additional light on the challenge involved in connectivity. We now turn our attention to connectivity facilitatation.

3 Facilitating Connectivity

In speculating the Society of Mind (SOM), Minsky states that [11]

"All natural languages, we are told, have much the same kinds of nouns, verbs, adjectives, and cases ... One possibility is that detailed syntactic restrictions are genetically encoded, directly, into our brains ... It would seem inevitable that some early high-level representations, developing in the first year or so, would be concerned with ... the 'things' that natural languages later represent as nouns. Here we indeed suspect generic prestructuring, within sensory systems designed to partition and aggregate their inputs into data for representing 'things.' Further, we would need systems with elementary operations for constructing, and comparing descriptions."

It is, of course, not our intent here to debate the speculation and evidences of SOM. The important point is that each local database can be perceived as genetically prestructured with certain syntactic restrictions, and designed to partition and aggregate data for representing "things." Further, we would need systems with elementary operations for constructing, and comparing descriptions. In this way, each local database is regarded as a "simple mind" with the "deep structures" [11] along with some means for manipulating them. The "simple mind" here, of course, refers to the "prestructured" local schema, and local DML is used to manipulate the "simple mind."

If we accept this analogy, then the *transient case-shift* mechanism [11] can also be applied where the most highly developed case has the best-developed description structure. Furthermore, Pappert's Principle [3] can be applied on top of these "simple-minds" to accumulate new knowledge and organize the existing knowledge into

---

(3) States that some of the most crucial steps in mental growth are based not simply on acquiring new skills, but on acquiring new administrative ways to use what one already knows [12].

different layers, with the top layer being an "intelligent global schema" capable of (a) activating the lower-level agents to retrieve the needed information from the local databases, and case-shift to the upper layers for the most highly developed cases; (b) reconciliating semantic conflicts among different agents through the Principle of Noncompromise[4]; and (c) transforming different sources of knowledge and information into composite-answers.

In light of this discussion, it is natural to apply frame-based representation [10] augmented with rule-based inference capabilities to facilitate connectivity.

Object-Oriented Approach

It is important to observe that certain *basic principles* can be applied to facilitate connectivity. For example, although tour-guides are idiocyncratic, each hotel has some rooms, and a room has some kind of facilities and services. As such, it can be represented as a composite-object with properties specialized in different granularities. These basic principles can serve as a reference of disparate databases, guiding the reconciliation.

The object-oriented approach enables us to encapsulate each database as an object. Messages can be sent among objects to obtain the information requested by a composite-query. In addition, object inheritance networks can be constructed to declare properties shared by different classes of objects. Active-values permit methods to be triggered (e.g., send a message to another object to obtain the interpretation of a hotel rating when information on rating is requested). Heuristic rules makes it convenient for describing flexible responses to a wide range of events, including situation where a unique global key identifier does not exist across

(4) States that the longer an internal conflict persists among an agent's subordinates, the weaker becomes that agent's status among its own competitors. If such internal problems aren't settled soon, other agents will take control and the agents formerly involved will be "dismissed" [12].

databases (e.g., tour-guide databases use different key identifiers for hotels). In this way, knowledge of object attributes, inheritance properties, and heuristic rules can be accumulated in the CIS-Executive to reconcile semantic differences.

There are many object-oriented languages, such as LOOPS, that offer excellent features such as *composite-objects* and *perspectives* which are very useful in facilitating connectivity [16]. Since we wish to experiment with various novel concepts involving connectivity to multiple databases, we implemented a specialized frame-based knowledge representation scheme and rule-based inference capability using COMMON-LISP, as discussed below.

Implementation Strategy

An Abstract Data Base Management System (ADBMS) was implemented as a CIS-Executive to integrate disparate databases for composite-answers. ADBMS is a higher-level conceptual DBMS which conceals the implementation details of the actual DBMSs from other objects in the community. It sends queries (via messages) to multiple databases (e.g., AAA. FODOR, and MASS) to access the local data dictionaries, retrieve information, convert formats, and integrate different views. Adding a new DBMS will not result in any change to the existing applications.

Also implemented was a set of commands which provide the basic features of an object-oriented language with extensions to simplify constraint and knowledge representation. Mechanisms are provided for interfaces with databases as well as building, relating, and showing objects. The functional relationship among ADBMS, database objects, and the actual DBMS is illustrated in Figure 5. The reader is referred to Levine [7] for a detailed description of the underlying Knowledge-Object REpresentation Language (KOREL). A simplified example is presented below.

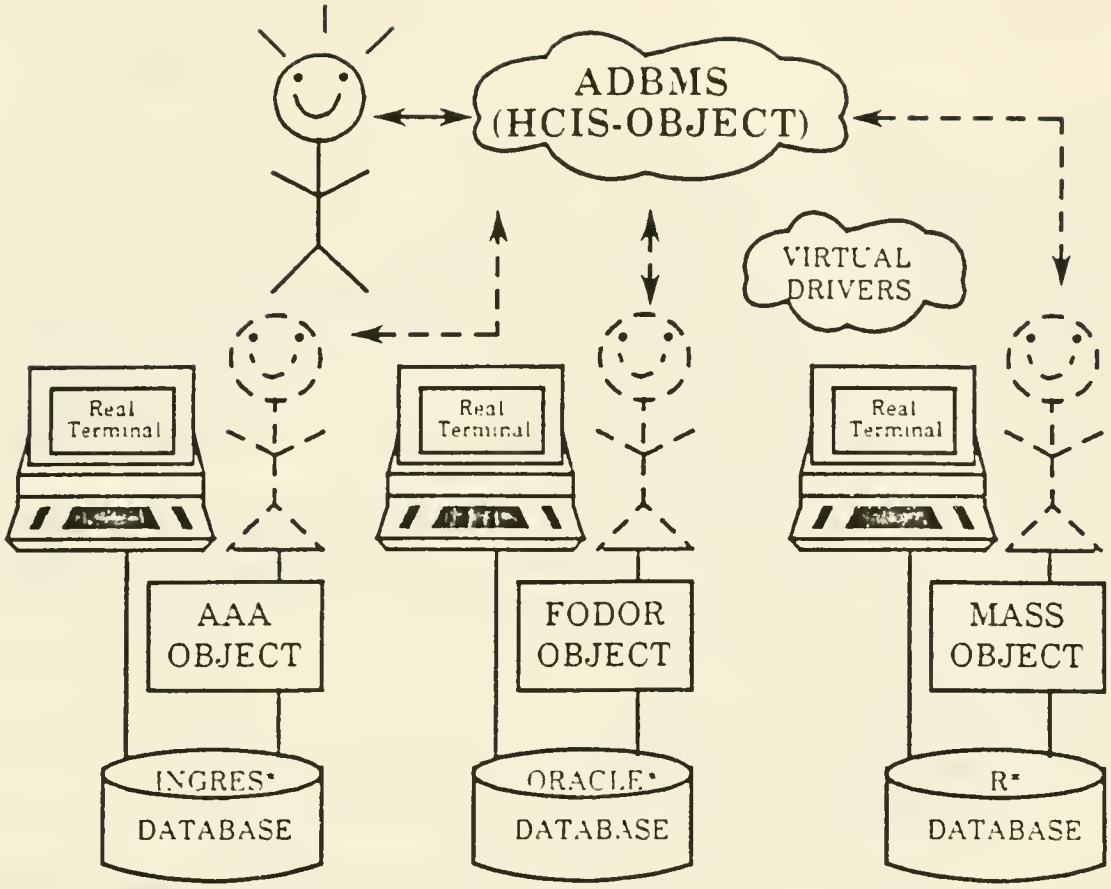

Figure 5 Functional Relationship Among ADBMS and the Actual DBMS

The Tour-Guide Case Revisited

Using the notation and the schemata in Figure 4, a Hotel-CIS (HCIS) object is realized in Figure 6 as a primitive ADBMS. Similarly, a partial representation of the FODOR object is shown in Figure 7. These objects are used below to present a simplified scenario of how to respond to the query "what are the facilities of Logan Airport Hilton" submitted by the user as discussed in the case study.

At the conceptual level, the command *"get-object 'HCIS 'facility '(Logan Airport Hilton)"* is executed to get a composite-answer of facilities from the HCIS object given the argument "Logan Airport Hilton."

At the HCIS level, the following commands are executed.

send-message 'FODOR 'all-facility '(Logan Airport Hilton)
send-message 'AAA 'all-facility '(Logan Airport Hilton)
send-message 'MASS 'all-facility '(Logan Airport Hilton)

```
(HCIS
    (NAME: (VALUE-TYPE string))
    (ADDRESS: (VALUE-TYPE string))
    (RATE-CODE: (VALUE-TYPE string))
    (LODGING-TYPE: (VALUE-TYPE string))
    (RATING: (VALUE-TYPE string))
    (#-OF-UNITS: (VALUE-TYPE integer))
    (PHONE# (VALUE-TYPE string)
        (MULTIPLE-VALUE-FUNCTION true))
    (DIRECTION: (VALUE-TYPE string))
    (FACILITY: (IF-NEEDED find-facility))
    (CREDIT-CARD (VALUE-TYPE string)
        (MULTIPLE-VALUE-FUNCTION true))
    (SEASON-RATE: (VALUE-TYPE string)
        (MULTIPLE-VALUE-FUNCTION true))
    (LOCATION: (VALUE-TYPE string))
    (COMMENT: (VALUE-TYPE: string))
    (PACKAGE: (VALUE-TYPE string))
    (SERVICE: (IF-NEEDED find-services)))
```

Figure 6 An Object Representation of the Composite-Model

```
(FODOR
    (MESSAGE: (VALUE all-facility))
    (FACILITY: (QUERY find-facility))
    (CATEGORY: (CHOICES super-deluxe,
                deluxe, expensive,
                moderate, inexpensive)
               (QUERY find-category)))
```

Figure 7 A Partial Representation of the FODOR Object

These messages trigger the AAA, FODOR, and MASS database objects to return all facilities that they "know of." After all the facilities have been returned to HCIS, it reconciles all the differences from the local results, formulate a composite-answer, and then return the composite-answer to IFEP for the user. In reconciliating the differences, heuristic rules are used to identify a hotel or resolve contradictory information.

We illustrate the FODOR case at the database object level. In response to the message from HCIS, the FODOR object executes the command "*get-object 'FODOR 'facility '(Logan Airport Hilton)*." This command triggers the following asynchronous events:

- Establish the necessary connection and send queries to the FODOR database to retrieve the facilities in FODOR-Facility shown in Figure 4.
- Execute the command "*get-object 'FODOR 'category '(Logan Airport Hilton)*" to establish the necessary connection and send queries to the FODOR database

to to retrieve the category of "Logan Airport Hilton." (The category turns out to be "expensive.")

- Execute the command "*send-message 'FODOR-CATEGORY 'expensive 'facility*" to infer additional facilities. The command triggers the FODOR-CATEGORY object to retrieve the facilities implied by the category "expensive" (i.e., bath or shower, TV and phone in room, heating, A/C, restaurant, and bar), and return the implied facilities to the FODOR object.
- Concatenate the facilities returned from the FODOR database and FODOR-CATEGORY; then return the concatenated results to HCIS.

Finally at the actual FODOR DBMS level, the subqueries are executed to retrieve the facilities in FODOR-Facility and the category in FODOR-Info given "Logan Airport Hilton."

The implementation of this prototype has provided us with experience and feedback on the effectiveness of using an object-oriented approach augmented with rule-based inference engine to handle semantic reconciliation, and an opportunity to see how we may devise a layered approach to organize the existing knowledge and accumulate new knowledge, as Pappert's Principle suggested. We conclude with the following remarks.

4 Concluding Remarks

The major problems that need to be overcome in order to integrate disparate databases for composite answers include contradiction, inconsistency, and ambiguity. We have presented an approach to resolve these problems through enhancing the semantic power of the database integration. This approach exploits concepts drawn from frame-based knowledge representation scheme and rule-based inferencing.

We are actively researching connectivity in the data engineering context. Features crucial to connectivity in CIS are being designed and implemented on an AT&T 3B2 machine under the UNIX environment. The enhanced version will be used to dial directly into multiple on-line DBMSs simultaneously to demonstrate the feasibility of a system capable of attaining connectivity in a real environment with satisfactory performance.

State-of-the-art approaches in formulating composite-answers are mostly application dependent and ad-hoc in nature. This research is intended to provide a theoretical foundation of connectivity in order to reconcile different assumptions and perspectives due to different mental models embedded in the different databases to be integrated.

References

1. Batini, C. Lenzerini, M. and Navathe, S.B. "A Comparative Analysis of Methodologies for Database Schema Integration," ACM Computing Surveys, Vol. 18, No. 4, December 1986, pp. 323 - 363.
2. Frank, W.F., Madnick, S.E., and Wang, Y.R. "A Conceptual Model for Integrated Autonomous Processing: An International Bank's Experience with Large Databases," to appear in the 8th Annual International Conference on Information Systems (ICIS), December 1987.
3. Goldhirsch, D., Landers, T., Rosenberg, R., and Yedwab, L. "MULTIBASE: System Administrator's Guide," Computer Corporation of America, November 1984.
4. Heimbigner, D. and Mcleod D. "A Federated Architecture for Information Management," ACM Transactions on Office Information Systems, Vol. 3, No. 3, July 1985, pp. 253-278.
5. Hewitt, C. E. Office Are Open Systems. ACM Transactions on Office Information Systems, Vol. 4, No. 3, (July 1986), pp. 271-287.
6. Lam, C.Y. and Madnick, S.E., Composite Information Systems - a new concept in information systems. CISR WP# 35, Sloan School of Management, MIT, May 1978.
7. Levine, S., "Interfacing Objects and Database," Master's Thesis, Electrical Engineering and Computer Science, MIT, May 1987.
8. Litwin, W. and Abdellatif, A. "Multidatabase Interoperability," Computer. December 1986, pp. 10-18.
9. Lyngbaek, P. and McLeod D., "An Approach to Object Sharing in Distributed Database Systems," 9th International Conf. on VLDB. October, 1983.
10. Minsky, M. "A Framework for Representing Knowledge," in Readings in Knowledge Representation, Bachman and Levesque (Eds.) Morgan Kaufman Publishers, 1985.
11. Minsky, M. "The Society Theory of Thinking," Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Cambrdige, Mass., August 22-25, 1977.
12. Minsky, M. The Society of mind, Simon and Schuster, New York, 1986
13. Madnick, S. E. and Wang, Y. R., Modeling the INFOPLEX database computer: a multiprocessor systems with unbalanced flows. Proceedings of the 6-th Advanced Database Symposium (August 1986), pp. 85-93.
14. Madnick, S.E. and Wang, Y.R. "Evolution Towards Strategic Applications of Databases Through Composite Information Systems," Working Paper #1866-87, Sloan School of Management, MIT, Submitted for publication.
15. Smith, J.M. et al. "MULTIBASE - Integrating Heterogeneous Distributed Database Systems," National Computer Conference 1981. pp. 487-499.
16. Stefik, M. and Bobrow, D.G. "Object-Oriented Programming: Themes and Variations," The AI Magazine. Winter 1986, Vol. 6, No. 4, pp. 40 - 62.