FACILITATING CONNECTIVITY IN COMPOSITE INFORMATION SYSTEMS

Y. Richard Wang

Stuart E. Madnick

January 1988 #WP 1975-88

To appear in Data Base

Facilitating Connectivity in Composite Information Systems

Y. Richard Wang*
Stuart E. Madnick
E53-320, Sloan School of Management
MIT, Cambridge, MA 02139
(617) 253-6612
(* Currently on leave from the University of Arizona, Tucson)

ABSTRACT Timely access to multiple disparate databases which were independently developed and administered to produce composite information has become increasingly critical for organizations to gain competitive advantage. However, many inter-database problems such as inconsistency, ambiguity, and contradiction remain unresolved.

This paper presents an approach for resolving these problems. The techniques employed in this approach include schema integration, inter-database tables, attribute subsetting, object hierarchies, and heuristic rules. Schema integration techniques resolve the incompatibilities among the databases at the schema level. Inter-database tables resolve the semantic inconsistency and concept granularity at the instance value level. The inter-database instance identification table identifies an instance across databases. Object hierarchies represent schemata as well as instances. Finally, heuristic rules are used to facilitate the construction of the inter-database instance identification table and the production of composite information.

KEY WORDS AND PHRASES: Distributed Database Management Systems, Data Administration, Database Administration, Database Integration, Logical Connectivity, Object-Oriented Programming, Organizational Information Systems, Strategic Information Systems.

ACKNOWLEDGEMENTS Work reported herein has been supported in part by the Department of Transportation's Transportation Systems Center, the U.S. Air Force, the Center for Management of Information at the University of Arizona, and Citibank. The authors also wish to acknowledge the valuable comments from the anonymous referees on an earlier version of this paper.

1. Introduction

Significant advances in the price, speed performance, capacity, and capabilities of new database technology have created a wide range of opportunities for business applications. These opportunities can be exploited to meet corporate strategic goals. 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) of organizational information systems that require disparate databases to work together. This category of information systems has been referred to as *Composite Information Systems* (CIS) [15, 16, 18, 19, 20].

A key benefit of CIS is to provide timely access to multiple disparate databases in concert to produce composite information. The process for obtaining this benefit is referred to as *connectivity* in this paper. Without connectivity, it is difficult, expensive, time-consuming, and error-producing to produce composite answers from information which may be stored in different databases located in different divisions of organizations.

Many problems such as inconsistency and contradiction among the disparate databases have been dealt with on an ad hoc basis. This paper presents an approach for resolving these problems through enhancing the semantic power of the database integration. The enhanced approach evolved from our observation that inter-database incompatibilities at the instance value level as well as those at the schema level must be resolved. The methods used in this approach include schema integration, Inter-Database Table (IDT), Inter-Database Instance Identification Table (IDIIT), object hierarchies, and heuristic rules.

Concepts and research background of CIS are presented in the remainder of this section. Section 2 presents a case study of tour-guide databases to exemplify issues involved in attaining connectivity. In section 3, a connectivity strategy is presented. Finally, concluding remarks appear in section 4.

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

The potential strategic importance of information technology (IT) is now an accepted fact [5, 6, 14]. It has also become increasingly clear that the identification of strategic applications alone does not result in success for an organization. A careful coordination from the domains of strategic applications, information technologies, and organizational structures must be made in order to attain success, as depicted in Figure 1. However, no established process or methodology is available for linking strategic applications to the other two domains [9, 19].

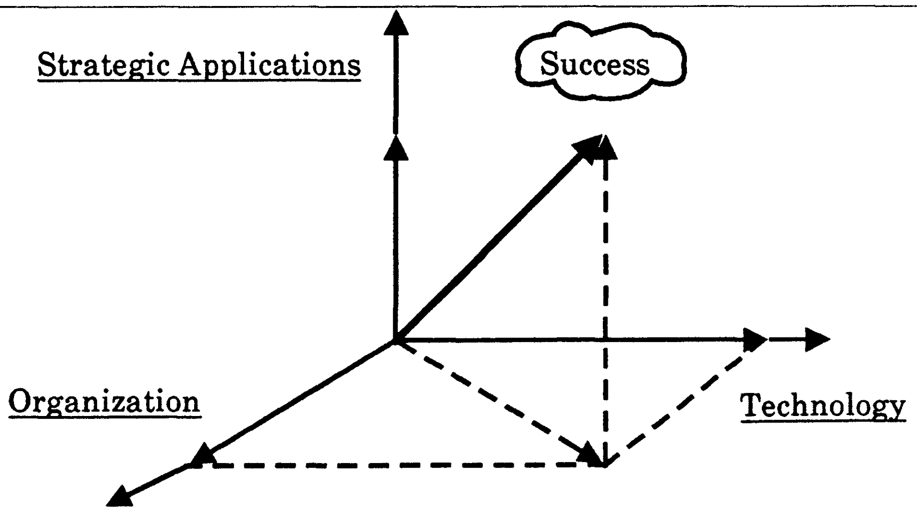

Figure 1 A Strategic Applications, Technology, and Organizational Research Initiative (SATORI)

An effective corporation is one that successfully reconciles the problems and opportunities of linking these three domains. It is important to recognize that no single pattern of interconnection among these three domains is likely to be consistently successful. Thus, one corporation may wish to lead from its technological domain and reconcile the other two domains accordingly. In contrast, another corporation may wish to develop its strategic applications from its product/market choice and develop its technological and organizational capabilities accordingly. The way that the corporation matches its internal capabilities with the external requirements determines its success in the marketplace. The primary research activities related to CIS are discussed below.

1.2 Related Work

The pioneering work on CIS began almost a decade ago [15]. Researchers in the information systems field have since evolved concepts such as inter-organizational information systems and distributed systems, which are summarized below.

Barrett and Konsynski [2] discussed concepts underlying the growth of inter-organizational information systems (IOS). A classification scheme was presented to examine issues of cost commitment, responsibility, and complexity of the operating environments. Barrett [1] further discussed a range of strategic options and IOS implementations. Their work represents a managerial perspective on the development and deployment of CIS.

In linking business and technology planning, Benson and Parker [4] argued that business planning should drive technology planning. Enterprise-Wide Information Management (EwIM) grids were proposed to enable practitioners as well academics to apply the EwIM tools of planning. Many of the IS planning tools such as Business Systems Planning (BSP) and Critical Success Factors (CSF) were mapped onto the grids. The work represents a trend towards articulating issues involved in business and IT at the planning level, eventually evolving into a methodology for linking strategic applications to appropriate IT and to the organizational context.

In the technical arena, much research has been conducted on the design of large capacity, cost-effective memory systems with rapid access time. Goyal and Agerwala [11] analyzed the performance of future shared storage systems. Madnick and Wang [17] modeled the INFOPLEX database computer in order to provide substantial performance improvements over conventional computers (e.g., up to 1000 fold increases in throughput) in information management, to support very large complex databases (e.g., over 100 billion bytes of structured data), and to insure extremely high reliability.

In parallel, the MULTIBASE research project at Computer Corporation of America [10] attempts to provide a uniform interface through a single query language and database schema to data in pre-existing, heterogeneous, distributed databases. The Federated Architecture [13] provides mechanisms for sharing data, for combining information from several components, and for coordinating activities among autonomous components via negotiation. Hewitt and De Jong at MIT [12] deal with highly parallel open systems. The underlying assumption of their research is that future IT applications will involve the interaction of subsystems that have been independently developed and administered at disparate geographical locations.

In the private sector, commercial database machines, such as Britton Lee's IDM 500 and Teradata's DBC 1012, have been introduced. Furthermore, homogeneous distributed database products such as INGRES* and SQL*STAR are now commercially available. It is conceivable that computation power approaching Cray 1 can be available on the desktop by the mid 1990's. Meanwhile, the window, mouse, and icon-based software coupled with rule-based techniques have provided the end user with easier and easier user interfaces to the computer-based information. Furthermore, commercial on-line databases such as Dow Jones^{TM} are increasingly accessible for up-to-date information.

The research results have created an opportunity for organizations to produce composite information that may be stored in different databases located in different divisions of organizations. Moreover, the increasingly available commercial products are important for implementing CIS with high return on investment, as illustrated below.

1.3 Strategic CIS Opportunities

Consider the following case study of a major international bank [9]. Currently, three separate database systems, shown in Figure 2, are used for cash management, loan management, and line of credit processing. Suppose a client requests that $100,000 be transferred from one account to another. If the client's cash balances in the funds transfer system can not cover the transaction, it will be rejected -- even though the client may have a $1,000,000 active line of credit! This rejection, besides

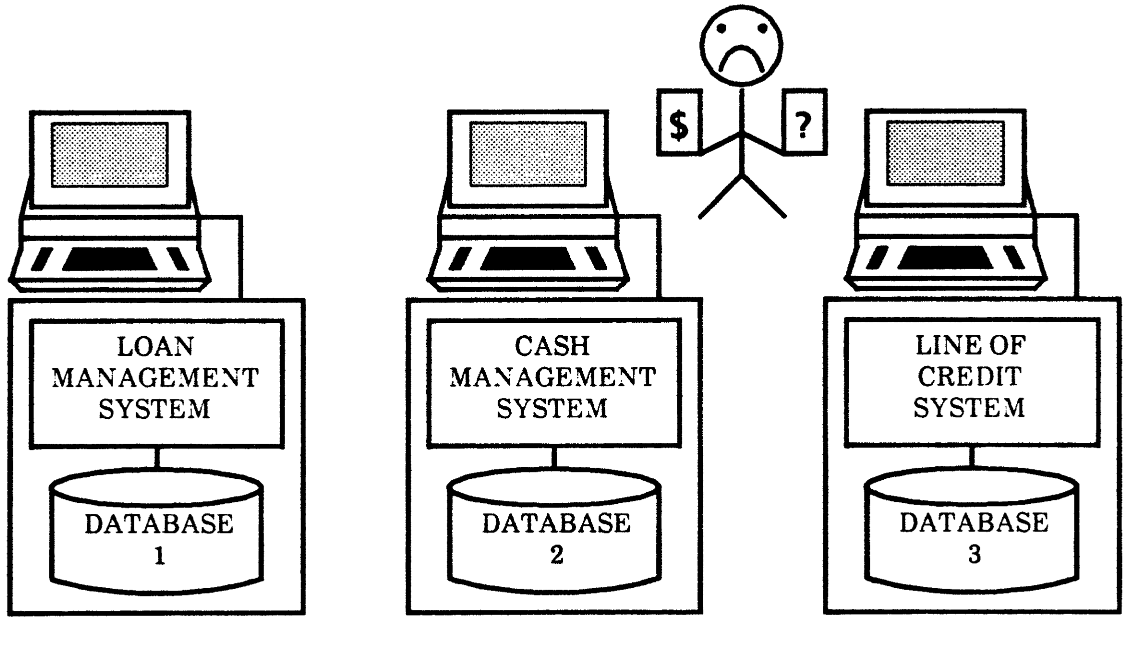

Figure 2 An Electronic Banking System Without Integration

being annoying and possibly embarrassing to the client, will require significant effort to correct by manually drawing on the line of credit to cover the transfer of funds.

If the bank can connect the three separate database systems together so that information is accessed in concert, and so that funds can be automatically drawn on the line of credit, then product differentiation will be achieved through the enhanced quality of service. Reprocessing costs will also be reduced because special manual intervention can be avoided.

Two levels of connectivity need to be considered in producing composite information: *physical connectivity* and *logical connectivity*. Physical connectivity refers to the process of actual communication among disparate databases. Although many issues need to be addressed in physical connectivity (e.g., bandwidths, security, availability, and reliability), we assume that adequate communication solutions are available. Our focus is on the semantic incompatibilities of databases, not on physical connectivity or on the DBMS used to implement the database. The process of resolving the semantic contradiction, inconsistency, and ambiguity that results from different assumptions made in disparate databases is referred to as logical connectivity. For brevity, connectivity hereafter refers to logical connectivity. A tour-guide case is presented below to illustrate issues involved in connectivity.

2. Tour-Guide Databases

Tour guides are easy to understand, abundant in data semantics, and representative of the situation involved in CIS. We chose tour guides in order to raise issues involved in resolving semantic incompatibilities in the delivery of timely, appropriate, and comprehensive composite information. Three tour guides are presented: AAA Tour Book for Massachusetts, 1987 (abbr. AAA hereinafter), FODOR's New England, 1987 (abbr. FODOR), and The Spirit of Massachusetts, 1987 (abbr. MASS). As discussed below, each tour guide contains somewhat different information and different degrees of detail or perspective on common information (e.g., average price of room, minimum and maximum room rates, and price of different types of rooms). To attain the most complete and comprehensive information, we would need to access all three tour guides. Let us suppose that AAA is implemented in INGRES*, FODOR in SQL*STAR, and MASS in R* by different organizations. Suppose also that we can access them in concert through computer networks to produce composite information such as price, location, and facility.

Interacting with a CIS front end processor, a tourist may wish to produce composite information about the facilities at the Logan Airport Hilton in Boston from all three tour guides. Let us see how we can formulate a composite answer for the question, "What are the facilities at the Logan Airport Hilton in Boston?" from the tour-guide databases with schemata shown in Figure 3.

2.1 Problems Encountered In Extracting Composite Information

Different queries need to be generated to access the relations in AAA, FODOR, and MASS to accumulate the facility data of the Logan Airport Hilton. In this process, it is necessary to realize that *amenity* in MASS is equivalent to *facility* in FODOR and AAA. In order to retrieve the data format of the facilities in Figure 3, the COLUMNS in the data dictionaries need to be accessed, as exemplified in Table 1. In addition, the amenity code in MASS has to be converted (e.g., 6 means pool).

The information that would be accumulated from that process is shown in Table 2 (except the entries with a "*"). In order to know that TV, A/C, phone, and heating are also available from FODOR, it is necessary to know that the Logan

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-Type, 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 3 Relational Schemata for AAA, FODOR, and MASS

Table 1 COLUMNS in the MASS Data Dictionary

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	Phone#	Char(13)
MASS-CC	CC	Char(2)
MASS-Amenity	Amenity-Code	Num(1)
MASS-Package	Package-Name	Char(40)
MASS-Package	Package-Descript	Char(80)

Airport Hilton is categorized as *expensive* by FODOR where *expensive* means, among other criteria, "bath or shower in each room, restaurants, TV, phone, attractive furnishings, heating, and A/C." Since the meaning of *expensive* is not stored as part of the relations, a procedure is needed to encode the information.

Table 2 Data for Logan Airport Hilton With Rating

AAA (Character 25 +)	FODOR (Character 30 +)	MASS (Numeric 1 +)
Parking lot		Free parking
C/TV	TV*	Cable TV
A/C	A/C*	Air Conditioning
Phones	Phone*	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
Cocktail	Bar	Lounge
Suites	Entertainment	Near public transportation
Smoke detectors	Heating*	Handicapped accessible

+ the data formats of the attributes.

* the facility inferred from the FODOR expensive category.

Many other semantic problems must also be resolved in order to formulate composite answers. Two examples are presented below to illustrate the complexity.

__Example 1__: How can one identify an instance across multiple databases?

A unique global key identifier may not always exist when multiple disparate databases are involved. For example, the names, addresses, and phone numbers 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 or 1-800-HILTONS

The identity of the lodging needs to be resolved in order to retrieve the facility data of "Logan Airport Hilton" across the three databases.

Example 2: How can one judge *credibility*?

Contradiction, granularity, and ambiguity are unavoidable when integrating disparate databases. It is necessary to make a "judgment call" when these issues arise. For example, AAA indicates that the Logan Airport Hilton has color TV without cable, but MASS reports that cable TV is available -- an apparent contradiction. A closer examination reveals that AAA has three categories for TV: C/TV for color TV, CATV for cable TV, and C/CATV for color cable TV; MASS indicates only if cable TV is available. Therefore, AAA is more detailed and may be assumed to be more credible in reporting TV information. The credibility knowledge needs to be incorporated if the contradiction is to be resolved.

If all the semantic problems can be solved, a composite answer for the facilities of the Logan Airport Hilton may be formulated as follows:

> "free parking; color TV without cable; air conditioning; phone in room; pool; airport transportation available; restaurant; non-smokers' room; and pets allowed. In addition, the following facilities have been reported: suites, smoke detectors, entertainment, cocktail, bar, lounge, near public transportation, and handicapped accessible."

2.2 Insights Gained From the Example

The tour-guide example revealed that two levels of incompatibilities, albeit not very distinct, need to be resolved: one at the schema level and the other at the instance value level. At the schema level, incompatibilities include synonyms, structural differences, and incompleteness.

- Type of lodging such as hotel, motel, and inn in AAA is referred to in MASS as type of facilities. They are synonyms at the attribute level (or entity level, depending on how they are modeled) since they refer to the same domain of values. The attributes "comment" in FODOR and "other" in MASS are also synonyms because both refer to the general comments given to a lodging. Similarly, amenity in MASS is equivalent to facility in AAA.
- Structural conflicts such as type conflicts and key conflicts are revealed. For example, "package" is a relation in MASS but an attribute in FODOR, causing a

type conflict. ID# is used in FODOR, but name is used in MASS instead as the primary keys, causing a key conflict.

- Incompleteness arises since each guide specializes in certain aspects of the problem domain. For example, AAA has a detailed rate relation while FODOR specializes in service and location.

At the instance value level, incompatibilities occur on a continuum, ranging from simple to complicated. In a simpler case, code conversion may suffice since a regular pattern may be available. For example, the amenity code 6 means pool in MASS, but the characters "pool" are used directly in AAA. This type of conversion can be easily made once the incompatibility is recognized. In a very complicated case, however, each instance value may be inconsistent, as exemplified by the lodging identification problem across the tour-guide databases (discussed in Example 1). The granularity and ambiguity of instance values may further complicate the problem. The following section presents a connectivity strategy to resolve these problems.

3. Connectivity Strategy

The incompatibilities revealed from the tour-guide example suggest that schema integration methodologies [3, 7, 8, 10, 13] can be effective in resolving problems at the schema level. Schema integration offers the CIS developer an opportunity to identify the syntax and semantic problems inherent in disparate databases. On the other hand, inter-database tables (IDT), inter-database instance identification tables (IDIIT), and knowledge-based techniques are used to resolve incompatibilities and ambiguities at the instance value level, as section 3.2 discusses. FODOR and MASS are used to illustrate the schema integration process below.

3.1 Resolving Incompatibilities at the Schema Level

Techniques used in the literature[1] show that many incompatibilities between FODOR and MASS can be revealed and resolved, as listed below.

- The FODOR-info and MASS-info relations are renamed "lodging."
- The ID# in FODOR is not used since it is unique only locally. Instead, the lodging name is used as the primary key to identify a lodging. As we will elaborate later, lodging identification across multiple databases is a central issue in attaining connectivity.
- The attributes "comment" in FODOR and "other" in MASS are merged as an attribute of lodging, renamed "comments."
- The attribute "package" in MASS is converted into an entity in the integrated schema.
- The attribute "location" in FODOR is carried over as an attribute of lodging.
- The attributes "facility type" (renamed lodging type) and "# of units" in MASS are carried over as attributes of lodging.
- The entity "CC" (credit card) in MASS is also carried over to the integrated schema, renamed "Credit Card."
- The entity "amenity" in MASS becomes the entity "facility" in the integrated schema.

In this way, the obvious name conflicts, structural differences, and incompleteness between FODOR and MASS are resolved. The new entities for lodging, package, credit card, and phone# are depicted in the *extended entity relation diagram* [3] shown in Figure 4. However, many more subtle incompatibilities remain unresolved, as discussed below.

----

[1] Batini, Lenzirini, and Navathe [1986] gave an example of schema integration to serve as the background of a comparative analysis of methodologies for schema integration. Elmasri, Larson, and Navathe [1987] presented schema integration algorithms for federated databases and logical database design. Many issues in schema integration regarding entity, attribute, and relation equivalence have also been discussed by many other researchers.For instance, in resolving conflicts in different schemata, Dayal and Hwang [1984] included naming conflicts, scale differences, structural differences, and differences in abstraction.

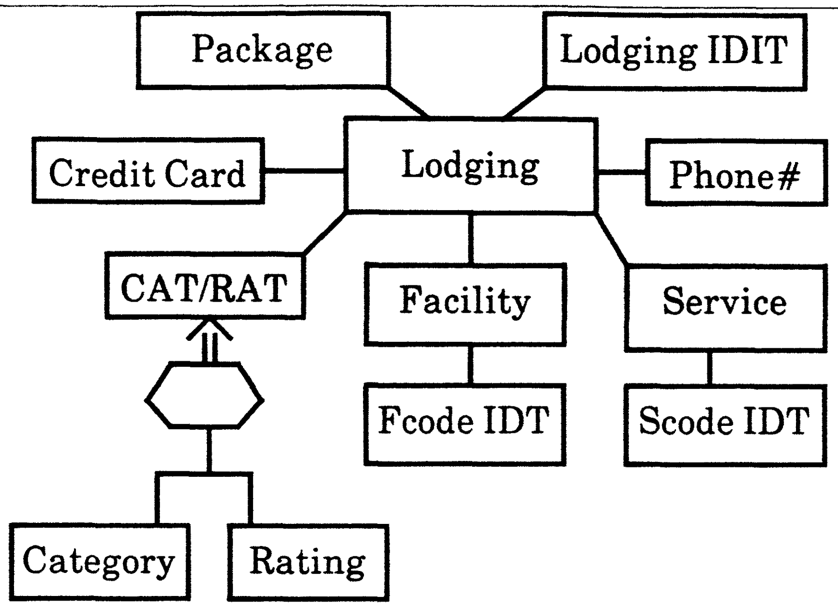

Figure 4 An Integrated Schema for FODOR and MASS

3.2 Resolving Incompatibilities at the Instance Value Level

Although the name conflict between amenity in MASS and facility in FODOR is resolved at the schema level, the problem is not solved yet at the instance level. For example, "outdoor pool" is used in FODOR, "pool" in MASS; similarly, "airport car available" is used in FODOR, "free transportation to/from airport" in MASS. This kind of problem can be avoided in the single database environment since the DB designer can predefine the domain values. In MASS, for example, the amenity code is used to encode the domain values (from 1 to 23, where 6 means pool); therefore, all the values in MASS for amenity have an exact interpretation. However, *there is a problem* when multiple databases are involved: in producing composite information, it is difficult for the computer to interpret the relationship between "outdoor pool" and "pool" or "airport car avail" and "free transportation to/from airport."

The phenomenon described above is not uncommon when multiple databases are involved. For each common attribute in two different databases, the domains need to be checked for their values. If the ranges are inconsistent, then an inter-database table (IDT) is created to reconcile the difference, as exemplified below.

3.2.1 Inter-Database Tables (IDT)

To resolve the facility differences in FODOR and MASS, a unique concept ID, concept group, and concept level are assigned to each concept. As shown in Table 3, the concept ID 101 is assigned to "A/C" in FODOR and "air conditioning" in MASS. Concepts with different degrees of granularity are assigned to the same concept group, but the more generic concept is assigned a higher number. For example, outdoor pool (103) and pool (104) are both assigned to the same concept group (3), but pool is assigned a higher number (2) than outdoor pool (1). In this way, the facility of FODOR and MASS are reconciled. Furthermore, such assignments provide a

Table 3 Inter-Database Table for Facilities

Concept Level	Concept group	Concept ID	Interpretation	Synonym
1	1	101	A/C	air conditioning
1	2	102	phone	telephone
1	3	103	outdoor pool	
2	3	104	pool	
1	4	105	color cable TV	C/CATV
1	4	106	cable TV	CATV
1	4	107	color TV w/o cable	C/TV
2	4	108	TV	
1	5	109	Free transportation to/from airport	
1	5	110	airport transport	airport car avail.
1	6	111	restaurant	dining rm
1	7	112	cocktail	
1	7	113	bar	
1	7	114	lounge	

mechanism to group and differentiate concepts. This mechanism is crucial for producing composite information.

Although the IDT provides a mechanism to group and differentiate concepts when a granularity problem arises, it does not help resolving a contradiction. Recall that AAA indicates "color TV **without cable**" as a facility, but MASS reports that "**cable** TV" is available. Since "cable TV" appears in both AAA and MASS, the same concept ID is used to encode the facility. As a result, the table parser can not detect the contradiction. One way to resolve the contradiction is to incorporate the judgment that *AAA* is more credible into the system, as shown in Table 4. The credit index for TV indicates that when in doubt, one should use the information retrieved from AAA.

Table 4 Credited Inter-Database Table For Facilities

Concept Level	Concept Group	Credit Index	Concept ID	AAA	FODOR	MASS
1	4	AAA	105	C/CATV		
1	4	AAA	106	cable TV		cable TV
1	4	AAA	107	color TV w/o cable		
2	4	AAA	108		TV	
1	6		111	dining rm	restaurants	restaurant
1	7	SAME	112	cocktail		
1	7	SAME	113		bar	
1	7	SAME	114			lounge

Note that the IDT also allows us to indicate that "dining rm" and "restaurant" are equivalent. In addition, it permits us to encode the judgment that "cocktail", "bar", and "lounge" are similar concepts (all with the same specificity, group, and credit index). The IDT for facility is depicted in Figure 4. Similarly, a service IDT is created for the entity service. We now turn our attention to another subtle incompatibility.

3.2.2 Converting Indecomposable Attributes

The attributes *category* in FODOR and *rating* in MASS were discovered to be neither disjoint nor equivalent[2]. The domains are {inexpensive, moderate, expensive, deluxe, super deluxe} for *category* and {$, $$, $$$, $$$$} for *rating* respectively. However, they do refer to something in common in terms of their role and structural identity. Although the literature has suggested that an attribute should be converted to an entity if it is represented as an entity in another schema (e.g., department is an attribute in one schema but an entity in the other), none has suggested, to our knowledge, how to integrate disjoint attributes such as *category* vs. *rating*. To produce the integrated schema as shown in Figure 4, we convert *category* and *rating* into entities, then create "CAT/RAT" as a generalized entity. Note that knowledge needs to be used to store the information for conversion purposes. We now turn our attention to an even more challenging incompatibility at the instance level -- the unique inter-database identifier problem.

3.2.3 Inter-Database Instance Identification Tables (IDIIT)

Recall that *Logan Airport Hilton* was reported as the name identifier for a particular lodging in AAA, *Hilton Inn at Logan* in FODOR, and *The Logan Airport Hilton* in MASS respectively, causing an identification problem. Such an instance level inconsistency can occur for each instance; on the other hand, in the facility attribute, the domain set has a limited number of values no matter how many

----

[2] Elmasri, Larson, and Navathe [1987] refined the characteristics of attributes and defined three types of attribute equivalences: (1) strong attribute equivalence; (2) weak attribute equivalence; and (3) disjoint equivalence.

instances exist in the databases. Note that this problem also occurs in the nonkey attributes such as address and phone numbers, which presume different values for different lodgings, causing potential inconsistency, ambiguity, and contradiction. The key uniqueness problem is more critical since it is used to identify the same lodging across multiple databases.

It is possible that a tax ID, which uniquely identifies a lodging, may be stored in FODOR and MASS . It may also be possible to find a combination of attributes to identify a lodging uniquely (e.g., tax ID, phone number, and zip code). If neither of the conditions exists but the problem can be confined with additional assumptions (such as only one phone for each lodging), then the problem is also reduced to one of the first two cases. If none of the above cases applies, then the attribute subsetting technique should be employed.

Attribute subsetting is a process for eliminating unrelated inter-database instances by comparing common attribute values. Instances that have a common attribute but have different attribute values are eliminated from the candidate set. For instance, if a target instance has a lodging type *hotel*, then instances in other databases which have lodging type *motel* are eliminated from the candidate set. Eventually a small set of instances in each of the databases is generated for the final identification.

The identification process can be done each time an instance needs to be identified. Alternatively, an *inter-database instance identifier table* (IDIIT) can be created whereby each lodging is assigned a unique inter-database ID, as shown in Table 5. Once the IDIIT is established, identifying a lodging across databases is a straightforward table look up. The trade-off is that IDIIT is proportional to the size of

Table 5 An Instance of IDIIT For Lodging

Inter-Database ID	AAA ID	FODOR ID	MASS ID
3456789876543	Logan Airport Hilton	Hilton Inn at Logan	The Logan Airport Hilton

the overall databases; it may be problematic if instance updating occurs frequently. The lodging IDIIT is also depicted in Figure 4.

We have presented several techniques to resolve the incompatibilities among the databases. It is interesting to note that artificial intelligence concepts, such as frames and rules, and the object-oriented approach provide a more expressive and general way of thinking about the problems and our solution techniques.

3.2.4 Knowledge-Based Techniques

The integrated schema shown in Figure 4 can be represented as frames. Many object-oriented languages (e.g., LOOPS) are now commercially available to implement frames and inheritance properties [21]. Our goal is to experiment with various novel concepts in a multi-process environment in which the direct access of multiple databases is possible. Therefore, we developed a specialized frame-based knowledge representation and rule-based inference prototype. We have dubbed it Knowledge Object REpresentation Language (KOREL) [16].

Figure 5 depicts part of the integrated schema represented in the KOREL notation. Each entity can be implemented as a frame with a set of slots. Each slot has one or more facets. For example, the entity lodging has slots for its attributes such as name, address, lodging type, #-of-units, direction, and comments. In addition, it has JOIN slots to link lodging with phone#, package, cat/rat, credit card, facility, and service frames. The JOIN slot has two facets: the join name and the join key. The generalized property is implemented through the subtype slot, as shown in the cat/rat frame, which has category and rating as its subtypes. Once the frames

```
(LODGING
    (NAME: (VALUE-TYPE string))
    (ADDRESS: (VALUE-TYPE string))
    (LODGING-TYPE: (VALUE-TYPE
    integer))
    (#-OF-UNITS: (VALUE-TYPE integer))
    (DIRECTION: (VALUE-TYPE string))
    (LOCATION: (VALUE-TYPE string))
    (COMMENTS: (VALUE-TYPE: string))
    (JOIN: (JOIN-NAME phone#)
           (JOIN-KEY name))
    (JOIN: (JOIN-NAME package)
           (JOIN-KEY name))
    (JOIN: (JOIN-NAME lodging-idit)
           (JOIN-KEY name))
    (JOIN: (JOIN-NAME cat/rat)
           (JOIN-KEY name))
    (JOIN: (JOIN-NAME credit-card)
           (JOIN-KEY name))
    (JOIN: (JOIN-NAME facility)
           (JOIN-KEY name))
    (JOIN: (JOIN-NAME service)
           (JOIN-KEY name)))
(CAT/RAT
    (NAME: (VALUE-TYPE string))
    (SUBTYPE: (category, rating))
    (JOIN: (JOIN-NAME lodging)
           (JOIN-KEY name)))
(PHONE#
    (NAME: (VALUE-TYPE string))
    (NUMBERS: (VALUE-TYPE string)
    (MULTIPLE-VALUE-FUNCTION true))
    (JOIN: (JOIN-NAME lodging)
           (JOIN-KEY name)))
(FACILITY
    (NAME: (VALUE-TYPE string))
    (FCODE: (VALUE-TYPE integer)
    (MULTIPLE-VALUE-FUNCTION true))
    (JOIN: (JOIN-NAME lodging)
           (JOIN-KEY name))
    (JOIN: (JOIN-NAME fcode-idt)
           (JOIN-KEY fcode)))
(CATEGORY
    (SUPERTYPE: (cat/rat)))
```

Figure 5 A Partial Representation of the Integrated Schema For FODOR and MASS in KOREL

are defined, KOREL commands can be used to invoke methods to produce composite information.

KOREL can also be used to represent the concept level, concept group, credibility, and other inheritance properties. Take the IDT for facility as an example. The issues there are how to represent synonyms, concepts, specificity, and credibility information, as shown in Table 3 and Table 4. An object hierarchy is created in Figure 6 to depict the concepts related to *facility*. The numbers from 101 to 114 denote the concepts identified in Table 3. A node "HI" is also created as a higher level concept for *cocktail*, *bar*, and *lounge*. Each object can be implemented as a KOREL frame. For example, *TV* (108) can be implemented as a frame that inherits properties from *facility* and *credibility* in AAA. It has slots for its concept ID (108),

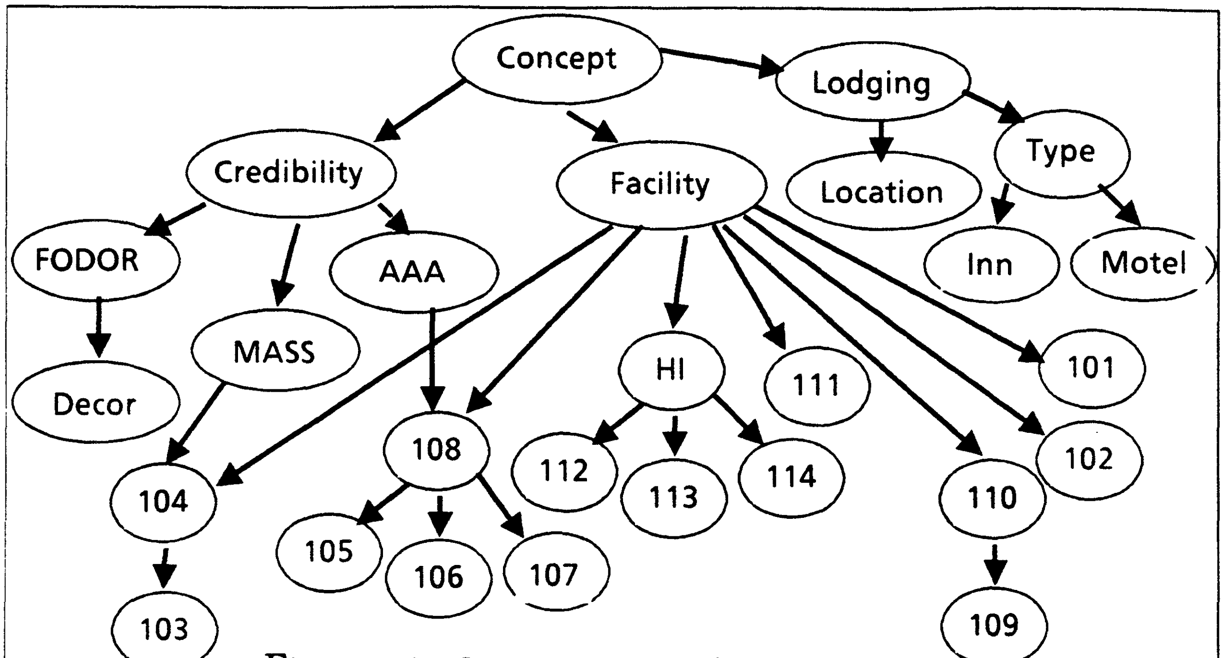

Figure 6 An Object Hierarchy for Facility

concept name (TV), and synonyms (e.g., television). The concept level and concept group are elegantly represented in the hierarchy.

It is interesting to observe the ramifications of giving MASS credibility for *pool* (104). Without the additional credibility information, *outdoor pool* (103) would be selected to formulate a composite answer because it is more specific than *pool.* With the new credibility information, an interesting situation is created in which the more specific information has less credibility (FODOR reported "outdoor pool" whereas MASS reported "pool"). A heuristic rule can be added to make the general judgment call. For instance, IF the concept level is higher but the source of data is more credible, THEN select the source of data.

Heuristic rules can also be employed to extract additional information unattainable before. In Figure 6, lodging information is included in the object hierarchy (which is not in Table 3 or Table 4 because lodging is not a facility). Conceivably, additional information about the facilities of a lodging is embedded in a lodging's location and its lodging type. For example, IF the lodging type is a *motel*, THEN it would be reasonable to encode a heuristic rule stating that free parking is available. Alternatively, IF a lodging's location is in the Boston Back Bay area (from zip code 02116), and the lodging is rated as $$$, THEN valet parking is available.

Another important application of the heuristic rules is in *attribute subsetting*. An instance may have many attributes to select for subsetting. The choice is domain specific and requires intimate knowledge of the application domain. In the lodging inter-database identification problem, for example, a lodging has many attributes. Furthermore, additional information for subsetting may also be available from other frames such as phone#, package, and credit card. How would the system know that it is useful to subset from lodging type and zip code instead of from comments or direction? Designing a good heuristic for attribute subsetting is a critical task. We are exploring general heuristics, which include rules such as "choose the attribute in the current set that has the maximum discriminating power." Our primary focus is on heuristics that are generalizable to various application domains.

We have illustrated frame-based representation, object hierarchy, and heuristic rules. The expressive power offered by knowledge-based techniques can be exploited in the implementation of a system to access multiple databases, as discussed below.

3.2.5 Prototype Implementation

An Abstract Data Base Management System (ADBMS) was implemented in KOREL as a CIS front end to access disparate databases for composite answers. ADBMS is a higher level conceptual DBMS that conceals the implementation details of the actual DBMSs from other objects in the community. It applies an integrated schema, as illustrated in Figure 4, of the local database schemata to implement the CIS front end. With the information from the integrated schema and the corresponding information from the local databases, it sends queries (via messages) to the local databases (e.g., AAA, FODOR, and MASS) to access the appropriate information.[3] Adding a new DBMS will not result in any change to the existing applications.

Also implemented was a set of commands. The commands 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 7. The reader is referred to Madnick and Wang [18] for a detailed example.

4. Concluding Remarks

As information technologies rapidly become available to society, a key issue for information systems researchers will be how to deliver timely, appropriate, and comprehensive information to the end user. To attain this information, one may have to extract information distributed throughout disparate databases within and/or across organizational boundaries. How to extract the appropriate information from these disparate databases efficiently, how to reconcile semantic differences among the databases so as to produce composite information, and how to deliver the

[3] Note that in the process of accessing the local databases, it is also necessary to translate a query in one general form into each particular format used by a local DBMS. This transformation would require very specific knowledge of the local DBMSs. Research conducted at the Computer Corporation of America on MULTIBASE [10] and more recently on PROBE has addressed the problem. A Global Data Manager (GDM) and Local Database Interfaces (LDI) were developed, for example, to perform the transformation from local databases to GDM. The reader is referred to [7, 10, 13] for a more detailed discussion of the issues involved in query transformation and modification in DBMS. Our research focuses on semantic reconciliation problems and instance identification problems in the contents of the databases.

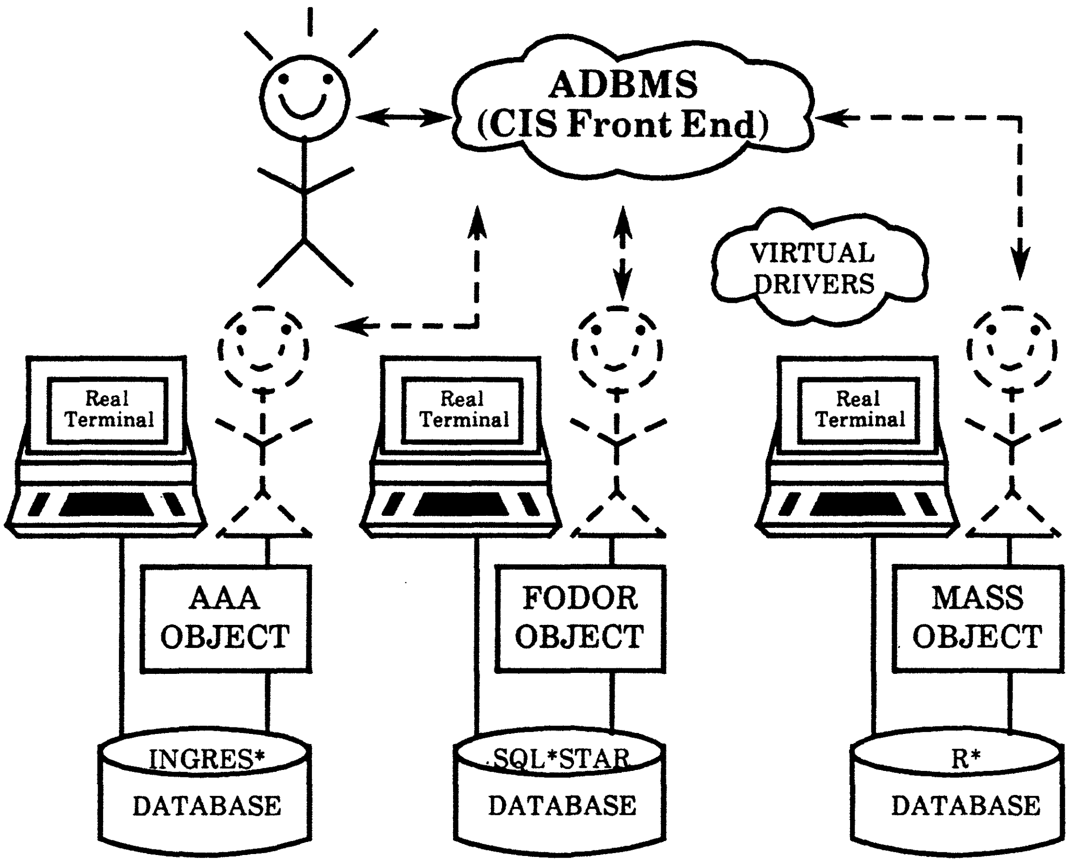

Figure 7 Functional Relationship Among ADBMS and the Actual DBMS

composite information to the user expediently are the issues that we have discussed in this paper.

We have presented a connectivity strategy based on schema integration, inter-database tables (IDT), inter-database instance identification tables (IDIIT), and knowledge-based techniques in order to resolve problems such as inconsistency, ambiguity, and contradiction; the resolution of those problems makes connectivity attainable. This research has provided a concrete step towards building a theoretical foundation of connectivity that reconciles the different assumptions and perspectives resulting from the different mental models embedded in the different databases being integrated.

References

1. Barrett, S. "Strategic Alternatives and Inter-Organizational Systems Implementations: An Overview," Journal of MIS, Vol. III, No. 3, Winter 1986-87, pp. 3-16.

2. Barrett, S., and Konsynski, B.K. "Inter-Organization Information Sharing Systems," MIS Quarterly, Special Issue 1982, pp. 93-104.

3. Batini, C. Lenzirini, 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.

4. Benson, R.J. and Parker, M. M. "Enterprise-Wide Information Management: An Introduction to the Concepts. IBM Los Angeles Scientific Center, G320-2768, May 1985.

5. Cash, J. I., and Konsynski, B.R. "IS Redraws Competitive Boundaries," Harvard Business Review, March-April 1985, 134-142.

6. Clemons, E.K. and McFarlan, F.W., "Telecom: Hook Up or Lose Out," Harvard Business Review, July-August, 1986.

7. Dayal, U. and Hwang, K. "View Definition and Generalization for Database Integration in Multidatabase System," IEEE Transactions on Software Engineering, Vol. SE-10, No. 6, November 1984, pp. 628-644.

8. Elmasri R., Larson J. and Navathe, S. "Schema Integration Algorithms for Federated Databases and Logical Database Design," Submitted for Publication, 1987.

9. 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," Proceedings of the 8th Annual International Conference on Information Systems (ICIS), December 1987, pp. 219-231.

10. Goldhirsch, D., Landers, T., Rosenberg, R., and Yedwab, L. "MULTIBASE: System Administrator's Guide," Computer Corporation of America, Cambridge, MA, November 1984.

11. Goyal, A. and Agerwala, T. "Performance Analysis of Future Shared Storage Systems," IBM Journal of Research and Development January 1984, p. 126-138.

12. Hewitt, C. E. Office Are Open Systems. ACM Transactions on Office Information Systems, Vol. 4, No. 3, July 1986, pp. 271-287.

13. 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.

14. Ives, B. and Learmonth, G.P., "The Information Systems as a Competitive Weapon," Communications of the ACM, Vol. 27(12), December 1984, pp. 1193-1201.

15. 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.

16. Levine, S., "Interfacing Objects and Database," Master's Thesis, Electrical Engineering and Computer Science, MIT, May 1987.

17. 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.

18. Madnick, S.E. and Wang, Y.R. Integrating Disparate Databases For Composite Answers. Proceedings of the Twenty-first Annual Hawaii International Conference on System Sciences, Vol. II, January 1988, pp.583-592.

19. Madnick, S.E. and Wang, Y.R. A Framework of Composite Information Systems for strategic advantage. Proceedings of the Twenty-first Annual Hawaii International Conference on System Sciences, Vol. III, January 1988, pp.35-43.

20. Madnick, S.E. and Wang, Y.R. "Evolution Towards Strategic Applications of Databases Through Composite Information Systems," To Appear in the Journal of Management Information Systems, Summer or Fall, 1988.

21. Stefik, M. and Bobrow, D.G. "Object-Oriented Programming: Themes and Variations," The AI Magazine, Vol. 6, No. 4, Winter 1986, pp. 40 - 62.