LOGICAL CONNECTIVITY:
Applications, Requirements, and an Architecture

Stuart E. Madnick
Y. Richard Wang

October, 1990 WP#2061-88
(Revision of August, 1988 WP) CIS-90-19

Logical Connectivity: Applications, Requirements, Architecture, and Research Agenda

Stuart E. Madnick Y. Richard Wang
Sloan School of Management, E53-320
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

This paper presents applications, requirements, an Information Technology (IT) architecture and a research agenda developed to facilitate connectivity among disparate information systems. The applications illustrate the benefits of connectivity as well as highlight problems encountered. These connectivity problems are organized into first- and second-order issues. The first-order issues include physical connectivity to heterogeneous databases in a multi-vendor environment. The second-order issues focus on logical connectivity, including schema-level integration, instance-level data semantics reconciliation, inter-database instance identification, and concept inferencing.

An IT architecture has been developed and an operational prototype implemented to address these issues. The salient features of this prototype include: (1) an object-oriented system which provides the required representation and reasoning capabilities, and (2) a three-layer software architecture which provides the required query processing and concept inferencing capabilities in the heterogeneous distributed database environment.

Based on these experiences, two new and important research issues have emerged which we refer to as the data source tagging problem and the source-receiver problem.

Key words and phrases
concept agents, concept inferencing, logical connectivity, heterogeneous databases, integration, information systems, objects, rules, query processing, systems development, tools.

1. Introduction

Advances in computer and communication technologies have provided significant opportunities for, and successful examples of, dramatically increased connectivity among information systems [8, 13, 21, 23, 29]. These opportunities, in turn, have enabled new forms of intra- and inter-organizational connectivity [2, 6, 11, 16, 28]. Meanwhile globalization, whereby the scope and presence of organizations expand beyond their traditional geographic boundaries, has propelled many organizations to capitalize on these increased connectivity opportunities. The inverse effect of globalization, world competition, is also on the rise as corporations expand through their globalization activities. These push-pull effects have been exemplified by the MIT Management School in its mission statement:

> "... because of the development of new information technologies, a world economy has emerged and is replacing what used to be an isolated national American economy. Today international competition primarily affects manufacturing, tomorrow it will become nearly universal. American financial services companies, for example, have come under acute international pressure. Foreign construction firms are seeking entry into the U.S. market. Foreign retailers are proving that they have the ability to buy or build retail chains that can take market share from American retailers. In the future few industries are going to benefit from a nationally protected market. Basic inputs, such as investment funds, technology, workers, and even management, are increasingly sourced on a world-wide rather than an American basis. The complex set of competitive and cooperative arrangements that firms can and do make with foreign firms is creating the need for managers with skills that used to be associated exclusively with diplomats. The American manager of the future will be an international manager even if he/she never leaves Kansas City." [32]

As a result of the interplay between the increased connectivity and globalization, many important applications in the 1990's will require access to and integration of multiple disparate databases both within and across organizational boundaries. This paper presents application examples, connectivity requirements, and an information technology architecture for increased connectivity among information systems. In addition, certain challenging research problems are emerging and are identified.

In this analysis, we assume that most existing systems cannot be easily changed -- either due to the cost, time, and complexity involved or because some of the systems are "owned" by autonomous organizations (e.g., an outside service or a customer's system). Thus, the integration must be accomplished through software that "surrounds" these existing systems rather than replacing them.

Section 2 discusses several real applications to illustrate the issues that are representative of a wide array of important connectivity applications. Section 3 investigates the requirements involved in attaining connectivity for a particularly challenging example involving both internal and external data sources. A Tool Kit for Composite Information Systems (CIS/TK) is presented in section 4 to support the requirements for increased connectivity. A CIS research agenda is introduced in Section 5. Finally, concluding remarks are made in section 6.

2. Example Applications

For an organization to gain strategic advantage [2, 4, 8] through increased connectivity, needs are manifest for at least three types of integration:

(1) Integration of internal services, such as the independent "current student" database admininistered by the Registrar's office and the "alumni" database administered by the Alumni office in a university, so that a user can appear to have a single database of <u>all</u> students, both past and present.

(2) Integration of external services, such as the Dataline service of Finsbury Data Services (which contains financial information on primarily European companies) and the Disclosure service of I.P. Sharp Associates, Inc. (which contains financial information on primarily American companies) so that a user can appear to have a single service that provides information on both European and American companies.

(3) Integration of the internal and external services.

As one example, Rockart [29] reported that Sun Corporation identified crude oil trading as perhaps the key business activity in the Refining and Marketing Division. Woody Roe was given the job of improving Sun's efforts in this area. He quickly realized the trading process was dispersed to a large number of groups located worldwide, each acting relatively independently. Some reported to the management of other Sun divisions. Roe envisioned a central trading room supported by information from Reuters and other trade data sources. Today coordinated, on-line trading is recognized by Sun executives as a major weapon in Sun's fight for increased revenue and profit in its very competitive industry.

Roe essentially created a system to provide timely access to multiple databases in concert to produce the desired composite information for the traders. We refer to the process of obtaining this benefit as *connectivity* [33, 34, 35]. Without connectivity, it is difficult, expensive, time-consuming, and error-prone to produce composite answers from information which is dispersed to a large number of groups located worldwide, each acting relatively independently.

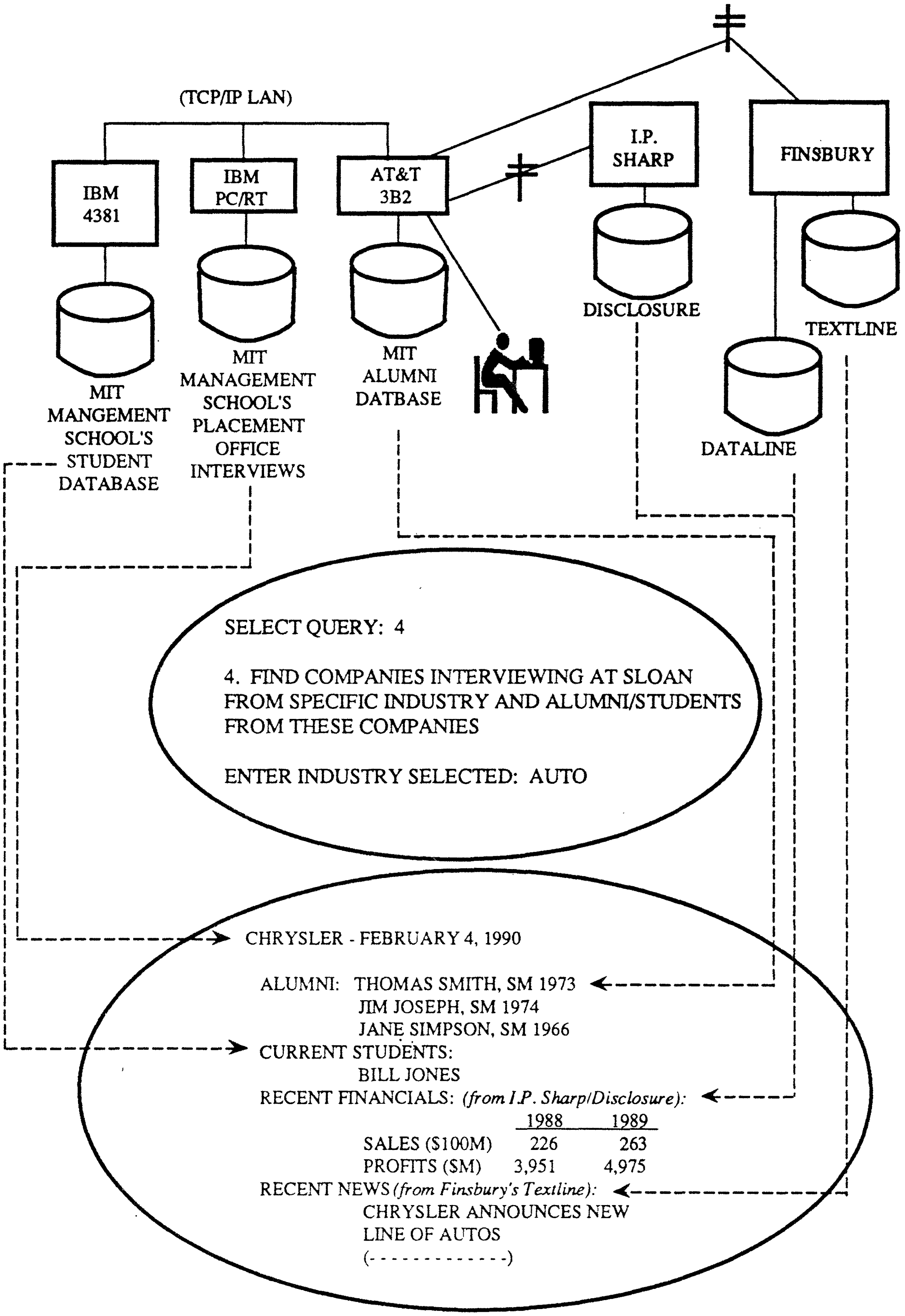
However, the real work for increased connectivity has just begun. There are many problems facing the design of integrated information networks, such as for financial dealing rooms*. In fact, attempts to solve some problems may actually create more serious and subtle problems. For example, the London Stock Exchange quotes most stocks in pence (*p*) but some stocks are quoted in pounds (£) [1£ = 100 *p*]. Although the printed newspaper stock listings indicate which stocks are £ quoted, this information is not transmitted on the exchange's real-time stock quote data feeds. This is not a problem for a typical London stock trader, since learning which stocks are £ quoted is part of that person's expected knowledge base. On the other hand, in the new world of global financial trading and the search for arbitrage opportunities, a trader located in New York or Tokyo may occasionally deal in London quoted stocks. These individuals are less likely to be familiar with all the idiosyncracies of other exchanges.

As a solution to this problem, Reuters has standardized the reporting procedures for its real-time stock quote data feeds so as to convert all London stock quotes into pence. However, in reality, many trading room systems get the same data delivered from multiple sources, e.g., via the Reuters network and the Telerate network, since there is only partial overlap of coverage. The Telerate service distributes London stock quotes unaltered (i.e., does not convert £ quoted stock into pence). Thus, a trader following stock on the London Exchange may see a sell offer at 12 via Telerate and a buy offer at 1200 via Reuters -- imagine the arbitrage possibilities, especially for an automated trading system! This kind of problem, encountered in a central dealing room system, is ubiquitous when attempt ing to increase connectivity via access to multiple disparate databases or data feeds in concert.

3. Analysis of Connectivity Problems for a Specific Application

As another real application, consider the Placement Assistant System (PAS), being developed for the MIT Sloan School of Management's Placement Office (see Figure 1). To accomplish its goals, this system must span six information systems in four organizations: (1) the **student database** and **the interview schedule database** are located in theSloan School; (2) the **alumni database** is located in the MIT alumni office; (3) *the recent corporate*

* Private communications with J. Hardman, Client Systems Design Manager, Reuters, London.



```
SELECT QUERY:  4

4.  FIND COMPANIES INTERVIEWING AT SLOAN
FROM SPECIFIC INDUSTRY AND ALUMNI/STUDENTS
FROM THESE COMPANIES

ENTER INDUSTRY SELECTED:  AUTO
```

```
CHRYSLER - FEBRUARY 4, 1990

  ALUMNI:  THOMAS SMITH, SM 1973
            JIM JOSEPH, SM 1974
            JANE SIMPSON, SM 1966
  CURRENT STUDENTS:
            BILL JONES
  RECENT FINANCIALS: (from I.P. Sharp/Disclosure):
                               1988      1989
             SALES ($100M)      226       263
             PROFITS ($M)     3,951     4,975
  RECENT NEWS (from Finsbury's Textline):
             CHRYSLER ANNOUNCES NEW
             LINE OF AUTOS
             (-------------)
```

Figure 1: Connectivity for the MIT Management School's Placement Office

news is obtained from Finsbury's **Textline database**; and (4) the *corporate financial information* is obtained from I.P. Sharp's **Disclosure database** and Finsbury's **Dataline database.**

A sample query for the system to handle would be to "find companies interviewing at Sloan"

- that are *auto manufacturers*,
- the *students* from these companies,
- the *alumni* from these companies,
- *financial information*, and
- recent *news* about these companies.

This information would be very valuable to a student interested in a position in the automobile industry. Current students from these companies can offer first-hand information. The alumni in these companies may be able to "put in a good word" on his behalf. Recent financial information indicates the economic environment at that company as well as providing information that may be helpful during an interview. Finally, recent news will keep the student abreast of what is going on in that company as well as to be well prepared for the interview with the recruiters.

3.1 Types of Connectivity Problems

Based on prior research efforts in view integration, database integration, and Composite Information Systems [1, 3, 7, 9, 10, 12, 14, 15, 19, 20, 25, 33], we have categorized connectivity problems into first- and second-order issues. The first-order issues, which have been the subject of most of the research on distributed heterogeneous database management systems, can be thought of as annoying and inconvenient problems caused by differences in physical connections and syntax of commands. Even after the first-order issues are solved, one must still face the second-order issues. The second-order issues, which are the foci of this paper, refer to the difficult problem of reconciling semantic differences between information provided from multiple sources and inferring information that is not explicitly represented in the underlying databases.

3.2 First-order Issues

The first-order issues are encountered immediately when attempting to provide access to and integration of multiple information resources [1, 7, 9, 10, 19]:

- multi-vendor machines (IBM PC/RT, IBM 4381, AT&T 3B2, etc.)
- physical connections (Ethernet, wide-area net, etc.)
- different database accesses (ORACLE/SQL, IBM's SQL/DS, flat files, menu-driven systems)
- information composition (formatting)

3.2.1. Multi-vendor machines and physical connections. The issue of multiple vendor machines and physical communications are commonplace whenever information resources are dispersed across functional or geographic groups, be they intra- or inter-organizational. For example, the MIT Management School's recruiting database is implemented in an *IBM PC/RT* computer whereas the MIT alumni database being used in this project is stored in an *AT&T 3B2* computer. Communication idiosyncrasies exist between different machines.

For example, to communicate with the AT&T 3B2 each message line should be transmitted full-duplex and be terminated with a New Line (NL) character. On the other hand, the I. P. Sharp service wants message lines transmitted half-duplex terminated with a Carriage Return (CR) / New Line (NL) sequence. While the Finsbury service needs message lines transmitted full-duplex terminated with just a Carriage Return (CR) character.

These differences would not be a problem if a user needed to access only one of these services. The terminal characteristics would merely need to be set appropriately once. But if multiple services need to be used, especially on a casual and/or infrequent basis by users with limited technical background or comfort, these peculiarities could be extremely distracting.

3.2.2 Different database accesses. Assuming that hardware idiosyncrasies and networking problems are resolved, the next hurdle is the idiosyncrasies of different databases. For example, the recruiting database is implemented using the ORACLE relational database management system and, thus accessed through SQL type queries; whereas I.P. Sharp's Disclosure and Finsbury's Dataline financial databases are accessed through menu driven interfaces. Different command sequences and the corresponding skill are required in order to obtain the information available from these various information resources.

A menu-driven interface is a very common and convenient way to provide information services to non-technical casual users, but even then there are still challenges. For example, in a menu-driven interface there is usually a way to terminate the current menu selection and revert to a higher level menu selection. To accomplish this in I. P. Sharp's Disclosure you need to type the message "BACK." On the other hand, to accomplish this same task on Finsbury's Dataline, you need to type the "\" (back slash) character. In neither system is the user reminded of the appropriate convention, it is assumed that the user has read the documentation and has remembered these conventions.

3.3 Second-order Issues

Suppose that one is able to resolve the above problems, the even more difficult information composition task abounds with second-order issues [3, 12, 33] such as:

- database navigation (where is the data for alumni *position*, base *salary*, *Ford* sales, *Fiat* sales,etc.)
- attribute naming (*company* attribute vs. *comp_name* attribute)
- simple domain value mapping ($, ¥, and £)
- instance identification problem (*IBM Corp* in one database vs. *IBM* in another database)

3.3.1 Database Navigation. Database navigation is needed in order to determine which database to access to get the required information. Furthermore, on a menu-driven database, e.g., Finsbury's Textline, it is important to know which menu path to use to retrieve the desired information. Similarly, in a relational database system, it is necessary to know in which tables the required data is located (e.g., alumni position, company name) so that appropriate SQL queries can be formulated.

3.3.2 Attribute Naming. Entity and attribute names may be termed differently among databases, for example the attribute *company* in the alumni database corresponds to the *compname* attribute in the placement database.

Another example is illustrated in Table 1 in which attribute names used in the Finsbury Dataline and the I. P. Sharp Disclosure databases are compared [27]. Certain companies, such as "Reuters Holdings PLC," happen to appear in both databases. Table 1 compares the information retrieved for "Reuters Holdings PLC" from both databases; furthermore, the attribute names that appeared in the Annual Report are also shown. Although the values are the same, the attribute named "sales" in Dataline corresponds to "net sales" in Disclosure which corresponds to "revenue" in the Annual Report.

Although these differences may be easy to handle for a human, it is a formidable challenge for the computer system to automatically resolve them. Sometimes, even humans may have difficulty realizing that "Minority Interest" in Disclosure corresponds to "Adjustments" in Dataline.

Table 1 can also be used to see the benefits of drawing upon multiple sources. Disclosure provides certain information (e.g., "Cost of Goods") not provided by Dataline. Conversely, Dataline provides certain information (e.g.,

Table 1: Differences in Attribute Names Used
(Information on "Reuters Holdings PLC" from Three Sources)

Pounds (000) 1987	Annual Report	Disclosure	Dataline
866,900	revenue	net sales (income statement)	sales
108,800	profit attributable to ordinary share holders	net income (income statement)	earned for ordinary
178,000	profit on ordinary activities before taxation	income before tax	pre-tax profits
800	minority interest	minority interest	adjustments
69,200	taxation on profit on ordinary activities	provision for income tax	pre-tax profit - profit after tax
276,500	tangible assets	net property, plant, & equipment	net fixed assets
8,300	investments	investments and advances to subs	investments
508,400	production and communication costs	costs of goods	??
358,500	(revenue) - (p'n & c'n costs)	gross profit	??
245,700	??	??	trading profit
115,300	short term investments	marketable securities	??
900	minority interests (on capital)	??	minority int. (capital and reserves)
29,900	stocks	inventories	??
127,900	(debtors) + ??	receivables	??
294,800	(current assets) - ??	current assets	??
-44,700	total net current assets	??	net current (liabilities) assets

"Net Current Assets") not provided by Disclosure. By merging both sources, we are able to attain a more complete picture than would have been available for either source singly.

3.3.3 Domain Value Mapping. In addition to the schema level integration, it is necessary to perform mapping at the instance level. For example, sales may be reported in $100 millions, but revenue in $millions. Furthermore, in a multi-national environment, financial data may be recorded in $, ¥, or £ depending on the subsidiary. If an American wanted to do a financial comparison (e.g., between Ford and Fiat), it would be helpful if the Disclosure data (for Ford) and Dataline data (for Fiat) could be matched up and if the Fiat financial information were automatically converted from lira to dollars.

3.3.4 Inter-database instance identification. The *instance identification* problem becomes critical when multiple independently developed and administered information systems are involved because different identifiers may be used in different databases. For example, Ford is referred to as "Ford Motor Co" in the alumni database, "The Ford Motor Company" in the placement database, "Ford Motor Co" in the Disclosure database, and "Ford Motor (USA)" in the Textline database. Textline actually uses company codes, "FRDMO" being the code for "Ford Motor (USA)." On the other hand, the company code "FRDMO" exists in Dataline but is defined to mean "Ford Motor Co Ltd" since Dataline does not (usually) have American companies. Thus, the exact same company code (in this case "FRDMO") refers to different companies depending upon which of the Finsbury databases (Textline or Dataline) is being used.

Two types of inter-database instance identification problems are being addressed in this research: common key analysis and attribute subsetting. Although these problems will be described separately, there are situations where both issues arise simultaneously.

3.3.4.1 Common Key Analysis. Table 2 shows an actual sample of company names extracted from the alumni and placement databases. Even though the order of the lists have been deliberately scrambled, humans would have little difficulty matching up the names from the two databases for these four companies.

Humans accomplish this task by using various rules, such as "&" and "and" usually mean the same thing, the postfix "& Co" may be omitted, and so on. If the appropriate knowledge, represented by these rules, could be captured, a system could perform this inter-database instance identification automatically.

3.3.4.2 Attribute Subsetting. In the more complicated cases, no common key identifiers are available for joining the data across databases for the same instance [25, 33]. As an example, let us consider a different application involving the two databases depicted in Table 3. Suppose that the professor for *Management Information Technology* (MIS 564) and *Communication and Connectivity* (MIS 579) had a database of students who take 564 and 579; while the Teaching Assistant for 564 had a database for just the 564 students. In preparing for final grading, the professor would like to know the T.A.'s opinion about Jane Murphy, an instance in his student database. Unfortunately, the T.A. is not available at the time.

As Table 3 shows, the two databases do not share a common key identifier for joining the data since the TA used the student's nickname as the key. Under this circumstance, the conventional database join technique is not applicable and human interaction would be required to identify the same instance across databases, i.e., matching Jane Murphy from the professor's database with one of the students in the T.A.'s database.

A moment of sharp observation would lead one to conclude that the human interaction involves the process of subsetting through common attributes to eliminate the unrelated candidate students followed by some heuristics to draw a conclusion. There is one explicit common attribute in the two database, i.e., performance. In addition, there is an implicit common attribute once we realize that sec564 in database #1 (which indicates whether the student attends the morning, A.M., or afternoon, P.M., session of MIS564) corresponds to section in database #2. By applying these two attributes, the candidate students that correspond to Jane are reduced from the entire database to 5 (i.e., those who attend the A.M. section of 564 with strong performance, as shown in the first five

Table 2: Example of Differences Among Connectivity Keys

Alumni Database	Placement Database
Air Products & Chemicals	American Management Systems, Inc.
Arthur Young	Allied-Signal Inc.
Allied Signal Corp.	Air Products and Chemicals
American Management Sys. Inc.	Arthur Young & Co.
• • •	• • •

Table 3: Example of Attribute Subsetting

Database #1 (Created by the Professor for MIS 564 and MIS 579)

Name*	564	579	Sec564	Age	Performance	Address
Jane Murphy	Yes	Yes	AM	19	Strong	Marblehead

Database #2 (Created by the TA for MIS 564)

Nickname*	Section	Performance	Sex	Major	Status	Trans	Evaluation
Happy	AM	Strong	F	MIS	UG	car	sharp cookie
Sneezy	AM	Strong	F	Fin	UG	train	coordinator
Dopey	AM	Strong	F	MIS	UG	bike	hacker
Sleepy	AM	Strong	M	MIS	UG	car	wild card
Doc	AM	Strong	F	MIS	G	car	tough cookie
Grumpy	AM	Weak	M	?	?	?	discard
Bashful	PM	Strong	M	MIS	G	walk	routine

rows of the T.A.'s database.)

Using the other attributes in these databases, plus auxiliary databases and inferencing rules, one may come to the conclusion that Jane Murphy is "*Happy*." The process goes as follows:

- Jane is 19 years old; therefore, the status is most likely "UG" (undergraduate) [this eliminates "*Doc*"].
- Using a database of typical male and female names, we can conclude that Jane Murphy is a female [this eliminates "*Sleepy*"].
- Jane lives in Marblehead. Using a distance database of locations in New England, we can determine that Marblehead is 27 miles from Cambridge and therefore, it is unlikely that the transportation type is bike [this eliminates "*Dopey*"].
- Jane takes 564 and 579 which are the core courses for the MIS major; therefore, it is more likely that Jane Murphy is majoring in MIS [this eliminates "*Sneezy*"].

Therefore, Jane Murphy is "*Happy*" who has been evaluated as being a "sharp cookie" by the T.A. Thus, even though only a few attributes are common to both databases, further comparisons can be made because of these additional relationships between the databases. Note that this analysis requires a combination of database information and rules (as will be discussed further in the Concept Agent section).

3.3.5 Concept Inferencing. Sometimes it is possible, or even necessary, to infer information that is not explicitly represented in the underlying data. As one example, let us return to our discussion of Table 1. In the Dataline system, the reports produced explicitly identify the currency (e.g., dollars, pounds, francs, lira). This is obviously necessary in Dataline because it contains information on companies from many different countries. The Disclosure system, on the other hand, does not explicitly identify the currency. This makes sense if one assumes that either: (1) all companies are US-based and use dollar currency or (2) any information on non-US companies has been converted to dollars. As Table 1 demonstrates for the case of "Reuters Holdings PLC," both of these assumptions are false. The casual user would have no easy way to realize that the information being provided in that case was in pounds, not in dollars!

Fortunately there are ways to handle this situation. For example, the currency can be inferred from the address information in Disclosure's Corporate Resume report. (For "Reuters Holdings PLC," the information "LONDON UNITED KINGDOM EC4P 4AJ" is provided for the attribute named "CITY").

Even more complex examples can arise by using multiple sources of information. For example, if one observes a $2 billion extraordinary asset loss in Schlumberger Ltd (Netherlands Antilles) from the Disclosure database and a corresponding $2 billion extraordinary asset gain in Schlumberger (France) from the Dataline database, one might infer that an asset had been transferred from one subsidiary to the other. (In actuality, it might not be quite so obvious if the currencies were different.)

Both the Disclosure and Dataline systems provide financial analysis ratios, such as "return on assets" (roa). Since these are ratios, there is no need to worry about what currency is used. However, if one requested the roa on Reuters Holding PLC from both systems, two different answers would be provided even though all of the actual data values supplied to both systems are the same, as seen in Figure 1 earlier. The reason is because Disclosure and Dataline define both "return" and "assets", as used in the formula "roa = return / assets", differently. We would like the system to not only note that the roa's are different but also explain why they are different (e.g., "The roa's are different because Disclosure uses an asset value of 294,800 and Dataline uses an asset value of -44,700. This is because ...").

The idea behind *concept inferencing* is to enable the system to acquire and store the knowledge so that it can perform such inferencing actions without human intervention.

4. Tool Kit for CIS

Confronted with these problems, we have found it effective to integrate the information sharing capability of DBMS technology and the knowledge processing power of AI technology [5, 17]. The *Tool Kit for Composite Information Systems* (CIS/TK) is a research prototype being developed at the MIT Sloan School of Management for providing such an integrated set of knowledge base and database system tools [1, 7, 15, 18, 24, 26, 31, 32]. It is implemented in the UNIX environment both to take advantage of its portability across disparate hardware and its multi-programming and communications capabilities to enable accessing multiple disparate remote databases in concert.

The primary design goals of CIS/TK is to move responsibility from the user to the system in the following three areas: (1) physical connection to remote databases; (2) DB navigation, attribute mapping, etc.; and (3) advanced logical connectivity issues. These goals are supported through its object-oriented language, information processing capabilities, and concept agents, as presented below.

4.1 Object-Oriented Language

The CIS/TK object-oriented language is based on an enhanced version of our Knowledge-Object Representation Language (KOREL) which facilitates an object-oriented and rule-based approach [18, 23, 25, 29]. This language provides three benefits: (1) it gives us the capability to evolve the code for experimenting and developing innovative concepts; (2) it provides the required knowledge representation and reasoning capabilities for knowledge-based processing in the heterogeneous distributed DBMS environment; and (3) it is very simple to interface with off-the-shelf software products (e.g., ORACLE and INFORMIX) and information services (e.g., I. P. Sharp and Finsbury) through the I/O redirection and piping capability inherent in the UNIX environment.

4.2 Query Processor Architecture

The CIS/TK query processor architecture (see Figure 2) is the key to information processing across multiple information resources. The architecture consists of an Application Query Processor (AQP), a Global Query Processor (GQP), and a Local Query Processor (LQP) to interface with the query command processor (e.g. DBMS) for each information resource in the CIS.

The AQP converts an application model query, defined by an application developer, into a sequence of global schema queries, passes them on to the GQP, and receives the results. The primary query processor is the GQP. It converts a global schema query into abstract local queries, sends them to the appropriate LQPs, and joins the results before passing them back to AQP. The GQP must know where to get the data, how to map global schema attribute names to the actual column names used in the individual databases, and how to join results from different tables. The LQP establishes the physical connection between the host and the appropriate remote machines where information is stored, transforms the abstract local query into the appropriate executable query commands for the remote system, sends the executable query commands to the actual processor, receives the results, and transforms the results to the standard GQP format.

Equally important to the CIS/TK information processing capabilities are the *global schema* and *application models* shown in Figure 2. The Global Schema is an integrated schema that models the data and relationships which are available in a set of underlying disparate databases. Thus, its sole concern is the *available* data from the underlying databases, and it may only partially reveal the richness of the underlying reality. On the other hand, the application model is best thought of as a mental model of a set of objects, which completely models their inter-relationships including derived data that may not be present in the underlying databases. Numerous application models might exist for different uses of the underlying data represented in the global schema.

The query processor architecture provides a mechanism for selecting data from the underlying databases modeled by the global schema and application model. The application model in turn provides the context for creating concept agents, the basic building blocks for concept inferencing.

4.3 Concept Agents

Concept agents assume a closed world view based on information in the application model for a CIS: data are obtained through the application model if not locally available in the concept agent; rules are constructed based on data and relationships modeled by the application model. A concept agent is composed of a *concept definition component*, a *concept processor* and a *message handler*.

The *message handler* is responsible for (1) activating the concept processor upon reception of a message requesting a concept agent to pursue its goal, (2) returning the results upon completion of the goal pursuance, and (3) requesting and receiving data from the AQP and other concept agents

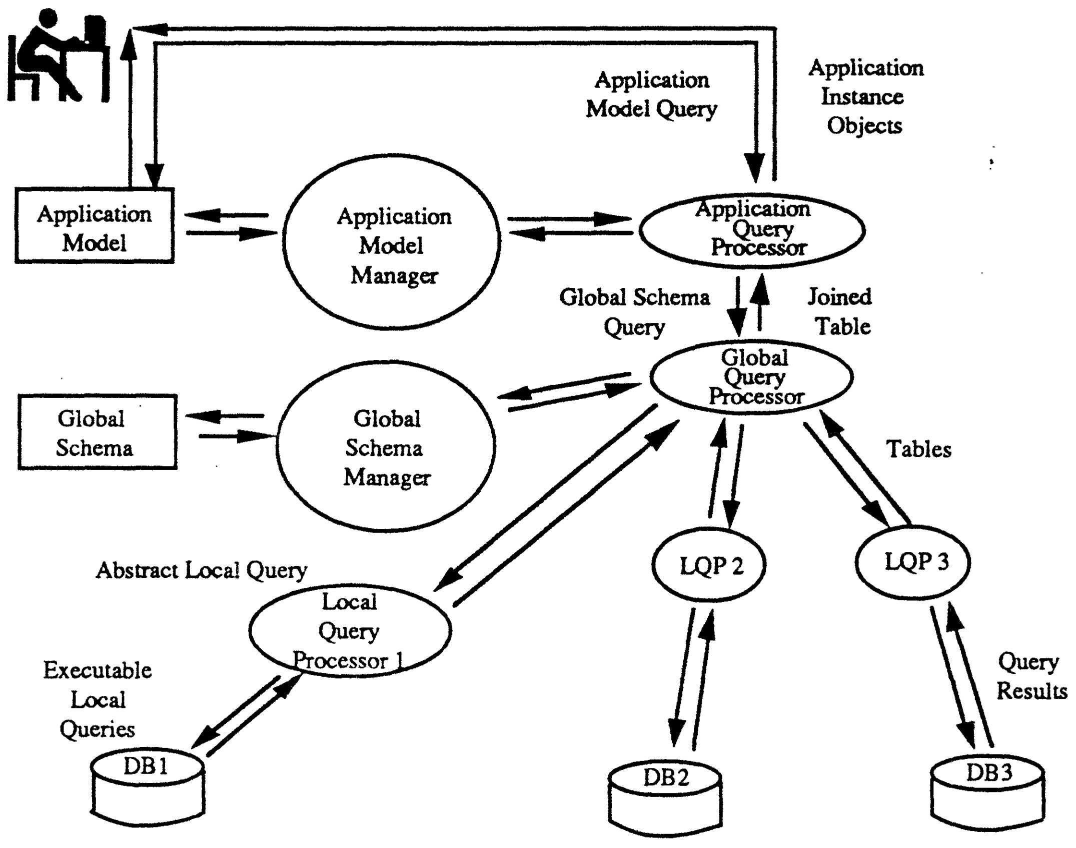


Figure 2: The CIS/TK Query Processor Architecture

on behalf of the concept processor.

All concept agents have the same *concept definition* structure. It defines the rules and data used by a concept agent as an object given an application model for a composite information system. As Figure 3 shows, the GOAL slot is where the concept processor places its assertion after processing its task. Each concept agent is capable of pursuing only a single, well-defined goal, or concept. Different perspectives of a goal can be accomplished as defined by the *concept processor*. The RULES slot contains all the rules that define the concept. The DATA-SLOTs are used to store local data and references to external data in the CIS/TK application model and the global schema. It has the following possible facets: (1) value, (2) procedure, (3) application-model, and (4) agent. The value facet stores the actual value if available. The procedure facet stores a procedure which computes the value for the slot. The app-model facet finds the value for the slot by referencing the entity called <entity> with the attribute called <attribute> of a CIS/TK application model called <model-name>. The agent facet finds the value for

```
GOAL:          <inferred value(s)>
RULES:         <all the rules that define the concept>
DATA-SLOT1:
        VALUE <values>
        PROCEDURE <procedure>
        APP-MODEL <model-name> <entity> <attribute>
        AGENT <agent-name>
DATA-SLOT2:
```

Figure 3: The Concept Definition Component of a Concept Agent

the slot by triggering the concept agent named <agent-name>.

The concept processor is responsible for firing the rules defining the concept agent, for triggering other concept agents to pursue subgoals, and for accessing data needed, internally and externally, in order to pursue its goal. It pursues the goal from three different perspectives: ***infer***, ***test-assertion***, and ***why***. *Infer* is used to infer all the values that satisfy the goal. *Test-assertion* is used when an assertion needs to be verified. *Why* is used when one wants to know how a concept agent arrives at the conclusion for either the *infer* and *test-assertion* cases.

5. CIS Research Agenda

In this section we examine our present research efforts in the development of new algorithms and approaches for composite informaiton systems.

5.1 Semantic Reconciliation Using Metadata (Source-Receiver Problem)

It has become increasingly important that methods be developed that explicitly consider the meaning of data used in information systems. For example, it is important that an application requiring financial data in francs does not receive data from a source that reports in another currency. This problem is a serious concern because the source meaning may change at any time; a source that once supplied financial data in francs might decide to change to reporting that data in European Common Units (ECUs).

To deal with this problem, the system must be able to represent data semantics and detect and automatically resolve conflicts in data semantics. At best, present systems permit an application to examine the data type definitions in the database schema, thus allowing for type checking within the application. But this limited capability does not allow a system to represent and examine detailed data semantics nor handle changing semantics.

We have examined the specification and use of metadata in a source-receiver model [30]. The source (database) supplies data used by the receiver (application). Using metadata we described a method for determining semantic reconciliation between a source and a receiver (i.e., whether the semantics of the data provided by the source is meaningful to the receiver).

The need to represent and manipulate data semantics or metadata is particularly important in composite information systems where data is taken from multiple disparate sources. To allow for greater local database autonomy, schema integration must be considered a dynamic problem. The global schema must be able to evolve to reflect changes in the structure and meaning of the underlying databases. If an application is affected by these changes, it must be alerted. As part of our research we are developing methods that use metadata to simplify schema integration while allowing for greater local database autonomy in an evolving heterogeneous database environment.

Methods for semantic reconciliation are described within a well-defined model for data semantics representation. This representation assumes that there are common primitive data types. From this base, for example, the currency of a trade price can be defined as the currency of the exchange where it was traded. Using this common language, sources can define the semantics of the data they supply and applications, using the same language, can define the semantic specification for required data.

Rather than a direct connection between the application and the database, the system includes a Database Metadata Dictionary component which contains knowledge about the semantics of the data and an Application Metadata Specification component which contains knowledge about the semantic requirements of the application. Semantic reconciliation is needed to determine if the data supplied by the database meets the semantic requirements of the application.

We have developed an algorithm that compares the semantic requirements of the application with the meaning of the data supplied by the source to determine if the source will supply meaningful data [30]. A similar algorithm can be used to determine if an application is receiving meaningful data from a set of component databases. In this case the data semantics of each database are examined to determine which sources might provide meaningful data. These methods can also be used to determine if an application can continue in the presence of changes in the component database semantics by making use of available conversion routines. Thus semantic reconciliation is a dynamic process which allows for component database semantic autonomy.

5.2 Polygen Model (Source Tagging Problem)

A typical objective of a distributed heterogeneous DBMS is that users must be able to access data without knowing where the data is located. In our field studies of actual needs, we have found that although the users want the simplicity of making a query as if it were a single large database, they also want the ability to know the source of each piece of data retrieved.

A polygen model has been developed to study heterogeneous database systems from this multiple (poly) source (gen) perspective [36]. It aims at addressing issues such as "where is the data from," "which intermediate data sources were used to arrive at that data," and "how source tags can be used for information composition and access charge purposes." In a complex environment with hundreds of databases, all of these issues are critical to their effective use.

The polygen model developed presents a precise characterization of the source tagging problem and a solution including a polygen algebra, a data-driven query translation mechanism, and the necessary and sufficient condition for source tagging. The polygen model is a direct extension of the relational model to the multiple database setting with source tagging capabilities, thus it enjoys all of the strengths of the traditional relational model. Knowing the data source enables us to interpret the data semantics more accurately, knowing the data source credibility enables us to resolve potential conflicts amongst the data retrieved from different sources, and knowing the cost of accessing a local database enables us to develop an access charge system.

A polygen domain is defined as a set of ordered triplets. Each triplet consists of three elements: a datum drawn from a simple domain in a local database (LD), a set of LDs denoting the local databases from which the datum originates, and a set of LDs denoting the intermediate local databases whose data led to the selection of the datum. A polygen relation p of degree n is a finite set of time-varying n-tuples, each n-tuple having the same set of attributes drawing values from the corresponding polygen domains.

We have developed precise definitions of a polygen algebra based on six orthogonal operators: project, cartesian product, restrict, union, difference, and coalesce. The first five are extensions of the traditional relational algebra operators, whereas coalesce is a special operator needed to support the polygen algebra. Other important operators needed to process a polygen query can be defined in terms of these six operators, such as outer natural primary join, outer natural total join, and merge. A query processing algorithm to implement a polygen algebra has also been developed.

6. Concluding Remarks

Recent business changes are both enabled by and are the driving forces towards *increased connectivity*. Some of the increasingly evident changes include (a) *globalization* of markets and product sourcing, requiring complex interconnectivity of systems, and (b) innovative information systems requiring a high level of strategy, technology, and cross-functional *integration* such as airline reservation systems locking in independent travel agents, direct order entry and order status inquiry between buyers and suppliers, and competitors linked in information networks for global securities trading.

In this paper, we have addressed these push-pull effects through application examples, connectivity requirements, a Tool Kit for Composite Information Systems (CIS/TK) and an emerging long-term research agenda. The CIS/TK ensemble is a unique and innovative system for delivering timely knowledge and information in an inter-organizational setting. An operational CIS/TK prototype has been developed at the MIT Sloan School of Management.

The complete prototype to be implemented will clearly demonstrate the feasibility of such an innovative concept. In the near term, we plan to extend the system through the following tasks: (1) design and implement facilities for more comprehensive inter-database instance identification and concept inferencing; (2) develop more efficient and more robust local query processors and communication servers; (3) demonstrate the feasibility of CIS/TK to interface and integrate with geographically and functionally disparate off-the-shelf products or customized systems, such as Finsbury's Textline and Dataline databases and I.P. Sharp's Disclosure and (4) incorporate the findings from our research on the source-receiver problem and the data source tagging problem. We believe that this effort will not only contribute to the academic research frontier but also benefit the business community in the foreseeable future.

References

1. Alford, M., and Wong, T.K., "The Implementation of A Tool Kit For Composite Information Systems, Version 1.0," Technical Report CIS-88-11, Sloan School of Management, MIT (August 1988).
2. Barrett S. "Strategic Alternatives and Inter-Organizational Systems Implementations: An Overview," *Journal of Management Information Systems*, (Winter 1986-87), Vol. 3, No. 3, pp. 3-16.
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. Benjamin, R.I., Rockart, J.F., Scott Morton, M.S., and Wyman, J. "Information technology: a strategic opportunity," *Sloan Management Review, Vol. 25*, No. 3, (Spring 1985), pp. 3-10.
5. Brodie, M. and Mylopoulos, J. (Ed.) On Knowledge Base Management Systems, Springer-Verlag (1986).
6. Cash, J. I., and Konsynski, B.R. "IS Redraws Competitive Boundaries," *Harvard Business Review*, (March-April 1985), pp.134-142.

7. Champlin, A., Interfacing Multiple Remote Databases in an Object-Oriented Framework. Bachelor's Thesis, Electrical Engineering and Computer Science, MIT, (May 1988).

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

9. 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.

10. Deen, S. M., Amin, R.R., and Taylor M.C. "Data integration in distributed databases," *IEEE Transactions on Software Engineering*, Vol. SE-13, No. 7, (July 1987) pp. 860-864.

11. Estrin, D. "Inter-Organizational Networks: Stringing Wires Across Administrative Boundaries," *Computer Networks and ISDN Systems 9* (1985), North-Holland.

12. Elmasri R., Larson J. and Navathe, S. "Schema Integration Algorithms for Federated Databases and Logical Database Design," Submitted for publication, (1987).

13. Frank, Madnick, and Wang, "A Conceptual Model for Integrated Autonomous Processing: An International Bank's Experience with Large Databases," *Proceedings of the 8th International Conference on Information Systems* (ICIS), (December, 1987).

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

15. Horton, D.C. An Object-Oriented Approach Towards Enhancing Logical Connectivity in a Distributed Database Environment. Master's Thesis, Sloan School of Management, MIT, (May 1988).

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

17. Kerschberg, L. Ed. Expert Database Systems, Proceedings from the First International Workshop. The Benjamin/Cummings Publishing Company (1986).

18. Levine, S., Interfacing Objects and Database. Master's Thesis, Electrical Engineering and Computer Science, MIT, (May 1987).

19. Litwin, W. and Abdellatif, A. "Multidatabase Interoperability," *IEEE Computer*, (December 1986).

20. Lyngbaek, P. and McLeod D. "An approach to object sharing in distributed database systems," *The Proceedings of the 9th International Conf. on VLDB*, (October, 1983).

21. Madnick (ed.) The Strategic Use of Information Technology. Oxford University Press, (1987).

22. Madnick and Wang, "Integrating Disparate Databases for Composite Answers," *Proceedings of the 21st Annual Hawaii International Conference on System Sciences*, (January 1988).

23. Madnick, S. and Wang, R. "Evolution Towards Strategic Applications of Data Bases Through Composite Information Systems," *Journal of MIS*, Vol. 5, No.2, (Fall 1988), pp. 5-22.

24. Manola, F. and Dayal, U. "PDM: An Object-Oriented Data Model," *Proceedings of the International Workshop on Object-Oriented Database Systems*. Pacific Grove, CA. (September 1986) pp. 18 - 25.

25. McCay, B. and Alford, M., "The Translation Facility in Composite Information Systems, Version 1.0," Technical Report CIS-88-10, Sloan School of Management, MIT (July 1988).

26. Ontologic Inc. Vbase Integrated Object System. 47 Manning Rd., Billerica, MA 01821 (November 1987).

27. Paget, M. An Object-Oriented Approach Towards Increased Connectivity for Globalization. Master's Thesis, Sloan School of Management, MIT, (December 1988).

28. Porter, M. and Millar, V.E., "How information gives you competitive advantages," *Harvard Business Review* (July-August 1985), pp. 149-160.

29. Rockart, J. "The Line Takes the Leadership: IS Management in a Wired Society," *Sloan Management Review*, *Vol. 29*, No. 4 (Spring 1988), pp. 57-64.

30. Siegel, M. and Madnick, S., "Schema Integration Using Metadata," Sloan School of Management, MIT, Cambridge, MA, WP # 3092-89 MS, (October 1989) and *1989 NSF Workshop on Heterogeneous Databases*,(December 1989).

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

32. Thurow, L. "The Mission of the MIT School of Management," *MIT Management*, Sloan School of Management, MIT, (Spring 1988) p. 31.

33. Wang, Y.R. and Madnick, S.E. "Facilitating Connectivity in Composite Information Systems," *ACM Data Base*, *Vol. 20*, No. 3, (Fall 1989).

34. Wang, Y.R. and Madnick, S.E. "Evolution Towards Strategic Applications of Databases Through Composite Information Systems," *Journal of Management Information Systems*, *Vol. 5*, No. 2, (Fall 1988).

35. Wang, Y.R. and Madnick, S.E. (ed.) *Connectivity Among Information Systems: Composite Information Systems Project, Volume 1*, MIT (September 1988).

36. Wang, R. and Madnick, S., "A Polygen Model for Heterogeneous Database Systems: The Source Tagging Perspective," *Proceedings of the 16th International Conference on Very Large Data Bases*, (August, 1990).