A POLYGEN DATA MODEL FOR DATA SOURCE TAGGING IN COMPOSITE INFORMATION SYSTEMS

Y. RICHARD WANG
STUART E. MADNICK

November 1989

WP # 3100-89 MSA
CIS-89-10

Composite Information Systems Laboratory
E53-320, Sloan School of Management
Massachusetts Institute of Technology
Cambridge, Mass. 02139
ATTN: Prof. Richard Wang
(617) 253-0442
Bitnet Address: rwang@sloan.mit.edu

ACKNOWLEDGEMENTS Work reported herein has been supported, in part, by MIT's International Financial Service Research Center, MIT's Center for Information Systems Research, and MIT's LCS Digital Library Project. The authors wish to thank Natasha Collins and Judith Mitiguy for helping prepare this manuscript and to Robert Goldberg and Allen Moulton for their comments on an earlier version of this paper.

A Polygen Data Model for Data Source Tagging in Composite Information Systems

ABSTRACT

Many important applications require access to and integration of multiple disparate distributed information systems both within and across organizational boundaries. These types of application systems have been referred to as *Composite Information Systems* (CIS). This paper studies such heterogeneous distributed systems from the "where" perspective.

Our experience in developing CIS for the business community indicates that most end-users wish to know the sources of their data. However, research prototypes and commercial products, to date, strive to encapsulate the heterogeneity of local databases in order to produce an illusion that all the information is obtained from a single data source. A *Polygen Data Model* (PDM) has been developed to tag data sources in CIS. Furthermore, we have established the strong and weak conditions for accurately tagging data in PDM. Based on the results, a polygen query processor is being implemented to tag data sources upon request of an end-user. When not needed, the processor behaves as a conventional relational query processor. It enables the user to apply his own judgment to the credibility of information dynamically, as well as to restrict information from a selected set of information systems. We believe that further research in this important area will not only contribute to the academic discipline but also benefit the business community in the foreseeable future.

ACRONYMS

AQP:	Application Query Processor
CIS:	Composite Information Systems
DST:	Data Source Tagging
ERM:	Entity Relationship Model
FDM:	Functional Data Model
LDB:	Local Data Base
LQP:	Local Query Processor
PDM:	Polygen Data Model
PQP:	Polygen Query Processor
RDM:	Relational Data Model
SFT:	Source Footprint Tagging

I. Introduction

Advances in computer and communication technologies have provided significant opportunities for dramatically increased connectivity among disparate information systems. At the same time, the scope and presence of organizations have expanded beyond their traditional geographic boundaries. This process of globalization has propelled many organizations to capitalize on increased connectivity opportunities. As a result, many important applications require access to and integration of multiple disparate distributed information systems both within and across organizational boundaries. These types of application systems have been referred to as *Composite Information Systems* (CIS) [25, 39, 47, 59-62, 64], *Federated Systems* [23, 29, 37], or *Integrated Information Systems.*

Many CIS research prototypes have been developed for composing information from multiple disparate systems, for example MULTIBASE in the United States [17, 27, 56], PRECI* in England [18, 19], and MRDSM in France [36]. To assess the capabilities of the state of the art industry practice, we have also surveyed more than forty commercial systems offering partial solutions to the distributed heterogeneous database problem, including Cincom's SUPRA, Metaphor's DIS, Oracle's SQL*loader, and TRW's Data Integration Engine (DIE) [28].

NEED FOR DATA SOURCE TAGGING

Our experience in working with the business community indicates that most end-users wish to know the data sources (e.g., "Source: Reuters' Newstext, December 6, 1989", or "Source: Finsbury's Dataline, 1990"). It enables them to apply their own judgement to the credibility of the information. For example, if Newstext and Dataline both report financial information about Merrill Lynch, but inconsistently, then an experienced financial analyst would reconcile the inconsistencies based on his[1] knowledge of the databases if the sources were disclosed[2]. A financial analyst may wish to incorporate data only from a restricted set of databases. Again, disclosing data sources helps.

To the best of our knowledge, the capability for an end-user to know the sources of data upon request has not been addressed in either research prototypes or commercial products. To date, those

---

1 or her. Masculine gender will be used hereinafter to denote both genders.

2 Whereas, a CIS would not achieve this on his behalf without a sophisticated knowledge base.

systems strive to encapsulate the heterogeneity of the underlying information systems in order to achieve transparency, namely producing an illusion that all the information is obtained from a single data source.

RESEARCH ISSUES

Data Source Tagging (DST) can be accomplished in a variety of ways. Since most organizations must deal with pre-existing systems that are controlled by autonomous divisions or separate organizations that are at best reluctant and at worst unwilling to change their systems [3, 6, 25, 38], it would be difficult to alter data in these systems to allow for DST. A more feasible approach is to tag the sources of data after they have been retrieved from each database.

In implementing DST, two related issues need to be addressed: (1) How does a CIS represent data? Which data model is used? (2) How does a CIS combine local data into a composite answer? Which data manipulation language is used?

Research in the CIS area has primarily adopted one of the following three data models [30, 50]: the Relational Data Model (RDM), the Functional Data Model (FDM), and the Entity Relationship Model (ERM). Examples of the RDM approach include MERMAID, PRECI* and MRDMS. FDM was used in MULTIBASE, and ERM in IISS. Each data model has merits for its intended purposes.

FDM is rich in semantics and is equipped with a powerful language - DAPLEX. ERM is also rich in semantics and is widely accepted as the leading database design tool. RDM lends itself to a simple structure and an elegant theoretical foundation. Its Relational Data Base Management Systems (RDBMS) dominate the database market today.

We selected RDM. Its data model and data manipulation languages are theoretically elegant and commercially accepted. Based on RDM, we developed *Polygen Data Model* (PDM)[3]. PDM extends the relational model to incorporate multiple (poly) sources (gen) of information. With PDM, we defined and proved the strong and weak conditions for the accuracy of DST in CIS. These results are being applied to guide the development of a cost-effective *Polygen Query Processor* (PQP).

---

3 In parallel, we are extending RDM to include Polygen Calculus and Polygen Completeness. The same mechanism presented in this paper is also being applied to develop Polygen ER-algebra, Polygen ER-calculus, and Polygen ER-completeness.

RESEARCH BACKGROUND

DST research is part of the CIS being developed at the Composite Information Systems Laboratory (CISL), Sloan School of Management, MIT. The current CIS prototype has access to three internal MIT databases (the Alumni Database, the Sloan Placement Database, and the Sloan Student Database) and three external commercial databases (Finsbury's Dataline and I.P. Sharp's Disclosure and Currency[4]). These databases provide breadth in data and examples of differences in style, accentuated somewhat by the different origins of each service [47]. For example, Finsbury is based in England, I.P. Sharp is based in Canada, and the MIT databases are based in the United States.

Following the ANSI/SPARC three-layer architecture, the CIS Application Query Processor (AQP) translates an end-user query into a composite query for the PQP, based on the user's application schema. The PQP in turn decomposes the composite query into a set of local queries, based on a *composite schema* of the relevant information in the local data bases (LDB), and routes them to the Local Query Processors (LQP). The details of the mapping and communication mechanisms between an LQP and its LDB is encapsulated in the LQP. To the PQP, each LQP manages a set of data available to it. Upon return from the LQPs, the retrieved data are further processed by the PQP in order to produce the desired composite information [60, 64]. Since there is a one-to-one correspondence between each LQP and its LDB, one can tag a datum either by an LQP name or its LDB name.

Many important problems need to be resolved in the CIS in order to provide a seamless solution to the end-user. The problems include data source tagging, schema integration [4, 23], inter-database instance matching [39, 62], domain mapping [61], and semantic reconciliation [39]. This paper focuses on the issues related to DST.

Section II presents the PDM. Section III illustrates the DST problem in CIS in the context of PDM. Section IV establishes the necessary and sufficient condition for the accuracy of DST in CIS using the PDM. Concluding remarks are made in section V.

---

4 Finsbury Data Services and I.P. Sharp Associates are both owned by Reuters Holdings PLC.

II. Polygen Data Model

Extending the relational model, we introduce the following basic definitions in PDM:

- A *polygen domain* is a set of ordered pairs. Each ordered pair consists of two elements: the first, t(d), is a value drawn from a simple domain in an LQP[5], and the second, t(s), is a set of DSTs where DST $\in$ {LQP names: for all LQPs in a CIS}. By definition, a *polygen domain is simple.*
- A *polygen relation p* of degree *n* is a finite set of *n*-tuples, each *n*-tuple having the same set of attributes drawing values from the corresponding polygen domains.
- A cell, *c*, in a polygen relation is an ordered pair of a datum and {DST} where {DST} $\subseteq$ {LQP names: for all LQPs in a CIS}. This type of tagging is defined as *DST-by-cell.*
- By definition, a *polygen relation* is *simple normal,* and polygen algebraic operations operate on polygen relations which are simple normal.
- Two polygen relations are *union-compatible* if their corresponding attributes are defined on the same polygen domain.
- A *polygen scheme* is a symbol *P* along with the *degree* of *P*, denoted deg(*P*). If *P* has degree *n*, then the *n* attributes of *P* are identified by their attribute identifiers or the numbers $1, \ldots, n$.
- A *polygen schema* is a sequence $\langle P_1, \ldots, P_N \rangle$ of polygen schemes.
- Let **I** be the set of all instances over a fixed schema of *N* polygen schemes. An *instance I*, $I \in \mathbf{I}$, of polygen schema $\langle P_1, \ldots, P_N \rangle$ is a sequence $\langle p_1, \ldots, p_N \rangle$, where for each $i = 1, \ldots, N$, p_i is a polygen relation of deg(P_i).

POLYGEN ALGEBRA

Building on Codd [11-15] and Klug [34], we define the set **E** of polygen algebraic expressions over a fixed schema as follows: If $e \in \mathbf{E}$ has degree *k*, then attrs(*e*) = $\{1, \ldots, k\}$. For each $e \in \mathbf{E}$ of degree *k* and each $I \in \mathbf{I}$, the *value* of *e* on *I*, denoted *e*(*I*), is a polygen relation of degree *k*.

(1) *Base polygen relations.* $P_i(I) = p_i$. $P_i \in \mathbf{E}$ for each P_i in the polygen schema, and deg(P_i) is as defined in the schema.

---

5 We assume that each LQP manages a set of data represented in RDM, as Section I discussed.

Let $t(d)$ denote the data portion and $t(s)$ the sources portion of tuple t. From the definition of *polygen domain,* we can further define *projection, Cartesian product, restriction, union,* and *difference* in the context of the PDM as follows:

(2) *Projection.* If $e \in \mathbf{E}$ and X is a sublist of attrs(e), then

$e[X](I) = \{ t'[X] : t'[X]_i = t_{x_i}$ if $t \in e(I) \wedge t(d)[X]$ is unique; $(t'(d)[X] = t_1(d)[X] \wedge t'(s)[X]_i = t_1(s)[X]_i \cup \ldots \cup t_k(s)[X]_i)$ if $t_1(d)[X] = \ldots = t_k(d)[X]\}$ where $t, t_1, \ldots t_k \in e(I)$.

$e[X] \in \mathbf{E}$ and deg($e[X]$) = length(X).

(3) *Cartesian product.* If $e_1 \in \mathbf{E}$ and $e_2 \in \mathbf{E}$, then

$(e_1 \times e_2)(I) = \{t_1 \circ t_2 : t_1 \in e_1(I)$ and $t_2 \in e_2(I)\}$, where $\circ$ denotes concatenation.

$(e_1 \times e_2) \in \mathbf{E}$, and deg $(e_1 \times e_2) =$ deg(e_1) + deg(e_2).

(4) *Restriction.* If $e \in \mathbf{E}$, $X \in$ attrs(e) and $Y \in$ attrs(e), then

$e[X\ \theta\ Y](I) = \{t : t \in e(I) \wedge (t(d)[X]\ \theta\ t(d)[Y])\}$ where θ is a binary relation $<, =, >, \geq, \neq,$ or $\leq$.

$e[X\ \theta\ Y] \in \mathbf{E}$ and deg($e[X\ \theta\ Y]$) = deg(e).

(5) *Union.* If $e_1 \in \mathbf{E}$, $e_2 \in \mathbf{E}$, and both have degree n, then

$(e_1 \cup e_2)(I) = \{ t' :$ $t' = t_1$ if $t_1(d) \in e_1(I) \wedge t_1(d) \notin e_2(I)$; $t' = t_2$ if $t_2(d) \notin e_1(I) \wedge t_2(d) \in e_2(I)$; $t'(d) = t_1(d) \wedge t'(s) = t_1(s) \cup t_2(s)$ if $t_1(d) = t_2(d)$ $t_1 \in e_1(I)$, $t_2 \in e_2(I)$. $\}$

$(e_1 \cup e_2) \in \mathbf{E}$ and deg($e_1 \cup e_2$) = n.

(6) *Difference.* If $e_1 \in \mathbf{E}$, $e_2 \in \mathbf{E}$, and both have degree n, then

$(e_1 - e_2)(I) = \{t : t \in e_1(I), t(d) \notin e_2(I)\}$.

$(e_1 - e_2) \in \mathbf{E}$ and deg($e_1 - e_2$) = n.

Other traditional operators can be defined in terms of the above operators. The most common are: *Selection, Join, Intersection,* and *Division.* Note that the PDM is *closed* under these operations, i.e., a PDM algebraic operation on PDM algebraic expression(s) yields a PDM algebraic expression. A critical issue here is that the closure property does not guarantee that the set {DST} will always tag exactly the sources of a datum. This critical issue will be addressed in Section IV.

III. Data Source Tagging in CIS

We illustrate query processing in CIS using three local schemata and a *composite schema.* This process helps shed light on data source tagging in CIS using the PDM.

SIMPLIFIED LOCAL SCHEMATA AND COMPOSITE SCHEMA

Suppose that the Alumni Database has the following relational schema:

ALUMNUS (AID#, ANAME, DEGREE, MAJOR)
CAREER (AID#, BNAME, POSITION)
BUSINESS (BNAME, INDUSTRY)

Each alumnus is uniquely identified through an alumnus identification number (AID#). Associated with each alumnus is a name, a degree, and a major. An alumnus may have positions in many businesses. Finally, a business is associated with an industry.

The Placement Database is illustrated below.

STUDENT (SID#, SNAME, GPA, MAJOR)
INTERVIEW (SID#, CNAME, JOB, LOCATION, SCHEDULE)
CORPORATION (CNAME, TRADE, CITY)

A student is uniquely identified by a student identifier number, and associated with a name, a GPA, and a major. A student may schedule interviews with many corporations for jobs at certain locations. Finally, a corporation is associated with a trade and is located in a city.

The Company Database is shown below.

FIRM (FNAME, CEO, INDUSTRY, HQ)
FINANCE (FNAME, YEAR, ASSETS, REVENUE, PROFIT)
OWNERSHIP (FNAME, TAXID#, SHARES)
INVESTOR (TAXID#, TYPE)

A firm has a name and a CEO. It is associated with an industry, and it is headquartered in a city. It discloses yearly financial information on assets, revenue, and profit. It is owned by many investors, each can be identified by a tax identifier number. Investors are institutional or private.

From the three local schemata, a *composite schema* can be constructed as follows:

CALUMNUS (AID#, ANAME, DEGREE, MAJOR)
CCAREER (AID#, ONAME, POSITION)
CORGANIZATION (ONAME, INDUSTRY, CEO, HQ)
CSTUDENT (SID#, SNAME, GPA, MAJOR)
CINTERVIEW (SID#, ONAME, JOB, LOCATION, SCHEDULE)
CFINANCE (ONAME, YEAR, ASSETS, REVENUE, PROFIT)
COWNERSHIP (ONAME, TAXID, SHARES)
CINVESTOR (TAXID#, TYPE)

The "C" before each relation name denotes that it is a *composite relation.* The mapping between the composite and local relations is a one-to-one correspondence. The only exception is the CORGANIZATION relation which merges the BUSINESS relation in the Alumni Database, the CORPORATION relation in the Placement Database, and the FIRM relation in the Company Database. Their attribute mapping relationship is shown below. In general, the mapping information between the composite and local relations and other pertinent metadata are managed by the PQP.

Composite Schema	Alumni Schema	Placement Schema	Company Schema
ONAME	BNAME	CNAME	FNAME
INDUSTRY	INDUSTRY	TRADE	INDUSTRY
HQ	---	CITY	HQ

EXAMPLE QUERY

In preparing a special report on the top ten graduate programs in Information Systems (IS) [ComputerWorld, October 30, 1989], Michael Sullivan-Trainor, a ComputerWorld staff, called MIT to get the names of CEO's who graduated from MIT's Sloan School of Management with an IS major.

For illustrative purposes, let us assume that the following SQL query[6] was submitted to the PQP based on the above *composite schema* in order to respond to Michael's request:

---

6 Instead of the simpler but less interesting approach of:
SELECT CEO
FROM CORGANIZATION, CALUMNUS
WHERE CEO = ANAME AND MAJOR = "IS" AND DEGREE = "MGT"

```
SELECT   CEO
FROM     CORGANIZATION, CALUMNUS
WHERE    CEO = ANAME
         AND  ONAME  IN
              (SELECT ONAME
              FROM  CCAREER
              WHERE  AID# IN
                        (SELECT AID#
                        FROM  CALUMNUS
                        WHERE MAJOR = "IS"  AND  DEGREE = "MGT"))
```

A corresponding relational algebraic expression for the SQL query is as follows:

(((((CALUMNUS [DEGREE = "MGT"]) [MAJOR = "IS"])
[AID#=AID#] CCAREER) [ONAME = ONAME] CORGANIZATION)
[CEO = ANAME]) [CEO]

Unlike the conventional RDBMS, schema mapping information is needed in CIS in order to map the algebraic expression from the *composite schema* to the local schemata. With the schema mapping information encoded in the CIS, the PQP can decompose the expression into an intermediate construct matrix, as shown in Table 1.

Table 1: An Intermediate Construct Matrix for the Example Query

R#	Operator	Algebraic Operation	Execution Location
R1	Selection	ALUMNUS [DEGREE="MGT"]	Alumni LQP
R2	Selection	R1 [MAJOR="IS"]	Alumni LQP
R3	Join	R2 [AID# =AID#] CAREER	Alumni LQP
R4	Join	R3 [BNAME=BNAME] BUSINESS	Alumni LQP
R5	Selection	FIRM	Company LQP
R6	Join	R4 [BNAME=FNAME] R5	PQP
R7	Restriction	R6 [CEO=ANAME]	PQP
R8	Projection	R7 [CEO]	PQP

Note that the first two rows (R1 and R2) indicate that a *selection* should be executed at the Alumni LQP[7], and a *join* for R3 and R4. Next, we see that R5 is a selection to be executed in the Company LQP. Finally, R6 is a *join*, R7 is a *restriction*, and R8 is a *projection* to be executed in the PQP. We are now in a position to tackle the DST problem.

---

7 In reality, the operation could be executed either in the LQP or the LDBMS, depending on the capability of the LDBMS.

DATA SOURCE TAGGING (DST)

Let us define DST to be accurate if the set of sources tagged to each datum denotes exactly the sources where the datum originates, for all the data in the resultant relation. For example, R8 contains a set of CEO's names from the Company Database, and only from the Company Database. In this case, DST is accurate if the set of sources tagged to each of the CEO's names is {Company Database}. The accuracy of DST will be defined more formally in Section IV. Note that in order to obtain R8, the PQP also used R4 which originates from the Alumni Database. In this case, the Alumni Database is defined as a *source footprint.* Issues involved in *Source Footprint Tagging* (SFT) are beyond the scope of this paper.

Granularity at the *relation* level may not be sufficient. Consider the *intermediate relation,* R6, which is the result of a *join* between R4 (from the Alumni LQP) and R5 (from the Company LQP). While R4 originates from the Alumni Database and R5 from the Company Database (the execution location is still at the LQP level), it is not clear where R6 originates (each tuple in R6 is a combination of data from the Alumni LQP and the Company LQP). Note also that sources may become indistinguishable once processed by the PQP. For example, the sources of R6 are unknown without additional information.

These observations manifest the approach we adopted in PDM for dealing with DST: (1) Tag the sources of data right after they are returned from the LQPs. (2) Tag the sources of data by cell. Table 2 illustrates how data and {DST} are combined, conceptually, in a *polygen relation* which corresponds to R3 in the example query - a *join* of ALUMNUS and CAREER on AID#.

Table 2: An Intermediate Polygen Relation Corresponding to R3 in the Example Query.

AID#	ANAME	DEGREE	MAJOR	POSITION	BNAME
(1, {AD})	(John Reed, {AD})	(SM, {AD})	(MGT, {AD})	(Chairman, {AD})	(Citicorp, {AD})
(2, {AD})	(Rich Wang, {AD})	(PhD, {AD})	(IS, {AD})	(Professor, {AD})	(MIT, {AD})
.	.	.	.	.	.

The first tuple in Table 2 denotes that the person with an alumnus identifier number 1 is John Reed. He has a Sloan Master (SM) degree with a major in management (MGT), and is chairman of

Citicorp. AD is tagged to each datum, denoting Alumni Database as the data source. Similarly, the second tuple denotes that Rich Wang's AID# is 2. He received a PhD degree with an IS major, and is currently a professor at MIT.

Table 2 contains only {AD} denoting the Alumni Database, because it represents a relation from the Alumni LQP. Once the polygen relation corresponding to R3 is further processed by the PQP, a cell may contain more than one DST. For instance, Table 3 illustrates a polygen relation corresponding to R6 in the example query.

Table 3: An Intermediate Polygen Relation Corresponding to R6 in the Example Query

AID#,	ANAME	DEGREE	MAJOR	ONAME	POSITION	INDUSTRY	CEO
(1, {AD})	(John Reed, {AD})	(SM, {AD})	(MGT, {AD})	(Citicorp, {AD, CD})	(Chairman, {AD})	(BANKING, {AD, CD})	(John Reed, {CD})
(2, {AD})	(Rich Wang, {AD})	(PhD, {AD})	(IS, {AD})	(Forea Inc., {AD, CD})	(Founder, {AD})	(COMPUTER, {AD, CD})	(Rich Wang, {CD})
.	.	.	.	.	.	.	.

In performing the *restriction* [CEO =ANAME] in R6, the PQP assumes that (John Reed, {CD}) is identical to (John Reed, {AD}) unless the user instructs the PQP to treat them differently.[8]

During *projection*, we may have an intermediate polygen relation whose cells contain identical data but different {DST}'s. For example, if a student wishes to know which company has IS job openings, and has a CEO who is an alumnus, then the PQP will produce an intermediate polygen relation as follows:

ONAME
(Citicorp, {AD, CD})
(Citicorp, {PD})
.

The first row, (Citicorp, {AD, CD}), indicates that Citicorp originates from both the Alumni Database (AD) and the Company Database (CD). The second row, (Citicorp, {PD}), indicates

8 The PQP must also be equipped with the capability to take on the data whose domains are not completely compatible. In the cases where sources are not compatible, additional domain mappings may be needed.

that the identical datum, Citicorp, has a different origin - the Placement Database (PD). In this case, *projection* merges the two attribute values, (Citicorp, {AD,CD}) and (Citicorp, {PD}) into one, namely (Citicorp, {AD, CD, PD}).

In the cases of *union* and *difference*, the data element follows *join* and the sources element follows *projection*. *Cartesian product* works as it does in the conventional relational algebra. The only difference is that it now takes polygen attributes instead of relational attributes. We have illustrated how data sources are tagged in CIS using the PDM. A critical question which remains unanswered is whether we can guarantee the accuracy of DST during CIS query processing using the PDM.

IV. Accuracy of DST in PDM

In order to establish the necessary and sufficient conditions for the accuracy of DST, we first introduce five definitions and a lemma.

(1) ***Strong Condition for DST Accuracy.*** Let cells(e[I]) denote the cells in e where $e \in$ E. Let c(d) denote the datum and c(s) denote {DST} where c=(datum, {DST}) and $c \in$ cells(e[I]). Define Data Sources Tagging to be *strongly accurate* if $\forall\, e \in$ E, the hypothesis, $\mathfrak{H}$, that $\forall\, c \in$ cells(e[I]), c(s) = {LQP name: every LQPs returns the datum to the PQP} is true.

(2) ***Weak Condition for DST Accuracy.*** Let cells(e[I]) denote the cells in e where $e \in$ E. Let c(d) denote the datum and c(s) denote {DST} where c=(datum, {DST}) and $c \in$ cells(e[I]). Define Data Sources Tagging to be *weakly accurate* if $\forall\, e \in$ E, the hypothesis, $\mathfrak{H}$, that $\forall\, c \in$ cells(e[I]), {LQP name: every LQPs returns the datum to the PQP} $\subseteq$ c(s) is true.

(3) Define *DST-by-relation* to be (e, {DST}) where e is an expression defined on some PDM.

(4) Define *DST-by-attribute* to be (x, {DST}) where $x \in$ attrs(e) and e is an expression defined on some PDM.

(5) Define *DST-by-tuple* to be (t, {DST}) where $t \in e(I)$ and e is is an expression defined on some PDM.

⟨Lemma 1⟩ ∃ four ways to DST: by cell, by tuple, by attribute, and by relation.

The granularity of a data object to be tagged in a PDM cannot be coarser than a polygen relation because a polygen relation is the basic unit of an algebraic operation. On the other hand, it

cannot be finer than a cell because a cell is the basic unit of a polygen relation. DSTs are deleted or merged by algebraic operators, all of them perform operations either by tuple (*Cartesian product, union, difference, and restriction*) or by attribute (*projection*). It follows that there are four ways to tag data sources in PDM: by cell, by tuple, by attribute, and by relation.Q.E.D.

⟨*Theorem*⟩ *DST is strongly accurate if and only if the granularity is DST-by-cell.*

⟨Proof⟩ Part 1: DST-by-cell ⇒ DST is strongly accurate.

Let e_1 denote a polygen relation that the PQP just received from an LQP, f denote an algebraic operation in PDM, $e_2 = f(e_1)$ for some $e_1 \in E$ if f is *projection, restriction;* $e_2 = f(e_1, e_1')$ for some $e_1 \in E$, $e_1' \in E$, if f is *Cartesian product, union, and difference.* Similarly, $e_{k+1} = f(e_k)$ for some $e_k \in E$ if f is *projection, restriction;* $e_{k+1} = f(e_k, e_k')$ for some $e_k \in E$, $e_k' \in E$, if f is *Cartesian product, union, and difference.*

We prove, by induction, that $\forall e \in E$, $\mathfrak{P}$ is true if the granularity is DST-by-cell.

By definition, $\forall e_1 \in E$, $\mathfrak{P}$ is true.

Let us assume that $\mathfrak{P}$ is true for $\forall e_k \in E$, and show that $\forall e_{k+1} \in E$, $\mathfrak{P}$ is also true.

For *projection*, $e_{k+1}(I) = e_k[X](I)$. Two cases need to be considered: (1) $t(d)[X]$ is unique, and (2) $t_1(d)[X] = \ldots = t_k(d)[X]$. In the first case, $c(s)$ remains the same for all $c \in t[X]$. In the second case, $t'(s)[X]_i(I) = t_1(s)[X]_i \cup \ldots \cup t_k(s)[X]_i$. Since $\mathfrak{P}$ is true for $t_1(s), \ldots, t_k(s)$, thus $\mathfrak{P}$ is true for $t'(s)[X]_i$. It follows that $\mathfrak{P}$ is true for $e_{k+1}(I) = e_k[X](I)$.

For *Cartesian product*, $e_{k+1}(I) = (e_k \times e'_k)(I) = \{t_1 \circ t_2 : t_1 \in e_k(I)$ and $t_2 \in e'_k(I)\}$, where $\circ$ denotes concatenation. For *difference*, $e_{k+1}(I) = (e_k - e'_k)(I) = \{t : t \in e_k(I), t(d) \notin e'_k(I)\}$. For *restriction*, $e_{k+1}(I) = e_k(X\ \theta\ Y)(I) = \{ t : t \in e_k(I) \wedge (t(d)[X]\ \theta\ t(d)[Y])\}$. Since $c(s)$ remains the same in *Cartesian product, difference,* and *restriction* for all $c \in$ cells$(e_k[I])$ and $c \in$ cells$(e'_k[I])$, it follows that $\mathfrak{P}$ is true for $e_{k+1}(I) = (e_k \times e'_k)(I)$, $e_{k+1}(I) = (e_k - e'_k)(I)$ and $e_{k+1}(I) = e_k(X\ \theta\ Y)(I)$.

For Union, $e_{k+1}(I) = (e_k \cup e'_k)(I) = \{ t' :$ $t' = t_1$ if $t_1(d) \in e_k(I) \wedge t_1(d) \in e'_k(I)$; $t' = t_2$ if $t_2(d) \notin e_k(I) \wedge t_2(d) \in e'_k(I)$; $t'(d) = t_1(d) \wedge t'(s) = t_1(s) \cup t_2(s)$ if $t_1(d) = t_2(d) \wedge t_1 \in e_k, t_2 \in e'_k \}$. By the same

token, $\mathfrak{P}$ is true for $e_{k+1}(I) = (e_k \cup e'_k)(I)$ Following the Principle of Mathematical Induction, we conclude that the proposition is true.

Part 2: DST is strongly accurate $\Rightarrow$ DST-by-cell.

Suppose that DST is strongly accurate and <u>not</u> by cell. It follows, by Lemma 1, that DST is by relation, by attribute, or by tuple. Suppose DST is *by relation.* Consider the *Cartesian product* of $(e_1, \{DST\}_1) \times (e_2, \{DST\}_2)$. By definition, the operation yields $\{t_1 \circ t_2 : t_1 \in e_1(I)$ and $t_2 \in e_2(I)$, where $\circ$ denotes concatenation, $source(t_1) = \{DST\}_1$, and $source(t_2) = \{DST\}_2\}$. However, the result cannot be expressed in the form of $(e, \{DST\})$. It follows that *DST by relation* is not sufficient. Suppose DST is *by attribute.* Consider *union.* By definition, $(t'(d) = t_1(d)) \wedge (t'(s) = t_1(s) \cup t_2(s))$ if $t_1(d) = t_2(d)$. However, the result cannot be expressed in the form of $(x, \{DST\})$. It follows that *DST by attribute* is not sufficient. Suppose DST is *by tuple.* Consider *Cartesian product.* By definition, the operation yields $\{t_1 \circ t_2 : t_1 \in e_1(I)$ and $t_2 \in e_2(I)\}$, where $\circ$ denotes concatenation, $source(t_1) = \{DST\}_1$, and $source(t_2) = \{DST\}_2$. However, the result cannot be expressed in the form of $(t, \{DST\})$. It follows that *DST by tuple* is not sufficient. Q.E.D.

It can be shown that the following corollaries are true.

⟨Corollary 1⟩ ∃ some PDM with DST-by-relation that satisfies the weak condition for DST accuracy.

⟨Corollary 2⟩ ∃ some PDM with DST-by-attribute that satisfies the weak condition for DST accuracy.

⟨Corollary 3⟩ ∃ some PDM with DST-by-tuple that satisfies the weak condition for DST accuracy.

V. Concluding Remarks

The Data Source Tagging (DST) research investigates the "***where*** " aspect of heterogeneous distributed systems research - an aspect that, to the best of our knowledge, has not been studied to date. In this paper, we presented a polygen data model (PDM) to frame the DST problem. In addition, we showed that DST is *strongly accurate* in PDM if and only if the granularity of source tagging is by cell. This result is being used to guide the development of a cost-effective polygen query processor.

PDM has provided us with a theoretical foundation to further investigate many other critical research issues in heterogeneous distributed systems research from the ***where*** perspective. For

example, a decision maker may want to know not only the sources of information in the resultant polygen relation but also the other sources that helped the composition of information in order to judge the credibility of the composite information - a type of problem that we call Source Footprints Tagging (SFT) problem. We believe that further research in this emerging area will not only contribute to the academic discipline but also benefit the business community in the foreseeable future.

Bibliography

1. Abiteboul, S. & Hull, R. IFO : A Formal Semantic Database Model. *ACM Transactions on Database System 12*, 4 (1987), 525-565.

2. Atzeni, P., & Chen, P. P. Completeness of Query Languages for the Entity-Relationship Model. In P.P. Chen ed., *Information Modeling and Analysis*, ER Institute, 1981. 111-124.

3. Barrett, S. Strategic Alternatives and Inter-Organizational Systems Implementations: An Overview. *Journal of Management Information System. 3*, 3 (Winter 1986-87), 3-16.

4. Batini, C., Lenzirini, M. & Navathe, S.B. A Comparative Analysis of Methodologies for Database Schema Integration. *ACM Computing Survey.* 18, 4 (December 1986) 323 - 364.

5. Brodie, M. & Mylopoulos, J. ed., *On Knowledge Base Management Systems*, Springer-Verlag, 1986.

6. Cash, J. I. & Konsynski, B. R. IS Redraws Competitive Boundaries. *Harvard Business Review*, (March-April 1985), 134-142.

7. Chen, P. P. The Entity-Relationship Model - Toward a Unified View of Data. *Transactions on Data Base Systems. 1*, 1 (1976), 166-193.

8. Chen, P. P. *A Preliminary Framework for Entity-Eelationship Model*, in P.P. Chen, ed., *Information Modeling and Analysis*, ER Institute, (1981), 19-28.

9. Chen, P. P. An algebra for a directional binary entity-relationship model. In *Proceedings of the 1st IEEE International Conference on Data Engineering*, (Los Angeles, CA, 1984) pp. 37-40.

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

11. Codd, E. F. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM. 13*, 6 (1970), 377-387.

12. Codd, E. F. Relational Completeness of Data Base Sublanguages, in R. Rustin, ed., *Data Base Systems*, Prentice-Hall, (1972), 65-96.

13. Codd, E. F. Extending the Database Relational Model to Capture More Meaning. *ACM Transactions on Database Systems*, 4 (4), (1979), 397-434.

14. Codd, E. F. Relational database: A practical foundation for productivity. In *Introduction to AI and Databases*, (1982), 60-68.

15. Codd, E. F. An evaluation scheme for database management systems that are claimed to be relational. Keynote address. *Proceedings of the International Conference of Data Engineering*, (1985), 720-729.

16. Dayal, U. & Hwang, K. View Definition and Generalization for Database Integration in Multidatabase System. *IEEE Transactions on Software Engineering, SE-10,* 6 (November 1984), 628-644.

17. Dayal, U., Hwang, H., Manola, F., Rosenthal, A. S., & Smith, J. M. Knowledge-oriented database management: Final technical report, Phase I. *Computer Corporation of America,* (Cambridge, MA., 1984).

18. Deen, S. M., Amin, R. R., & Taylor M. C. Data Integration in Distributed Databases. *IEEE Transactions on Software Engineering, SE-13,* 7 (July 1987), 860-864.

19. Deen, S. M., Amin, R. R., & Taylor, M. C. Implementation of a Prototype for PRECI*. *Computer Journal, 30,* 2 (1987b), 157-162.

20. DeMichiel, L. G. Performing operations over mismatched domains. In *Proceedings of the Fifth International Conference on Data Engineering* (Los Angeles, CA., February 1989).

21. dos Santos, C. S., Neuhold, E. J., & Furtado, A. L. A Data Type Approach to the Entity-Relationship Model. In P. P. Chen, ed., *Entity-Relationship Approach to Systems Analysis and Design,* North-Holland Publishing Co., 1980, 103-119.

22. Elmasri, R. & Wiederhold, G. GORDAS: A Formal High-Level Query Language for the Entity-Relationship Model. In P. P. Chen, ed., *Information Modeling and Analysis,* ER Institute, 1981, 49-72.

23. Elmasri R., Larson J. & Navathe, S. Schema Integration Algorithms for Federated Databases and Logical Database Design. Submitted for Publication (1987).

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

25. Frank, Madnick, & Wang. A conceptual model for integrated autonomous processing: an international bank's experience with large databases. In *Proceedings of the 8th International Conference on Information Systems* (ICIS, December, 1987).

26. Godes, D. B. Use of heterogeneous data sources: three case studies. In *WP # CIS-89-02.* (Sloan School of Management, MIT, Cambridge, MA., 1989). CISL Project.

27. Goldhirsch, D., Landers, T., Rosenberg, R. & Yedwab, L. MULTIBASE: System Administrator's Guide. *Computer Corporation of America.* (November 1984).

28. Gupta, A. An architectural comparison of contemporary approaches and products for integrating heterogeneous informations. In *WP # CIS-89-12.* (Sloan School of Management, MIT, Cambridge, MA., 1989). CISL Project.

29. Heimbigner, D. & McLeod, D. A Federated Architecture for Information Management. *ACM Transactions on Office Information Systems, 3,* 3 (1985) 253-278.

30. Hull, R. & King, R. Semantic Database Modeling: Survey, Applications, and Research Issues. *ACM Computing Surveys, 19,* 3 (September 1987), 201-259

31. Hwang, H. & Dayal, U. Using the entity-relationship model for implementing multi-model database systems. In P. Chen, ed., *Entity Relationship Approach to Information Modeling and Analysis* (California, E. R. Institute). 237-258.

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

33. Kerschberg, L., ed. Expert Database Systems, *Proceedings from the First International Workshop.* The Benjamin/Cummings Publishing Company 1986.

34. Klug, A., Equivalence of Relational Algebra and Relational Calculus Query Languages Having Aggregate Functions. *Journal of the Association for Computing Machinery. 29,* 3 (July, 1982), 699-717.

35. Lien, Y. E., Shopiro, J. E., & Tsur, S., DSIS - A database system with interrelational semantics. In *Proceedings of the 7th International VLDB Conference* (1981). 465-477.

36. Litwin, W. & Abdellatif, A. Multidatabase Interoperability. *IEEE Computer* (December 1986), 10-18.

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

38. Madnick, S., ed., *The Strategic Use of Information Technology.* Oxford University Press, 1987.

39. Madnick & Wang, Integrating disparate databases for composite answers. In *Proceedings of the 21st Annual Hawaii International Conference on System Sciences* (January, 1988).

40. Manola, F. & Dayal, U. PDM: An object-oriented data model. In *Proceedings of the International Workshop on Object-Oriented Database Systems* (Pacific Grove, CA., September, 1986), 18-25.

41. Manola, F. Distributed object management technology. In *Technical Memorandum, GTE Laboratories, TM-0014-06-88-165* (1988).

42. Manola, F. Applications of object-oriented database technology in knowledge-based integrated information systems. In paper prepared for the *CRAI School on Recent Techniques for Integrating Heterogeneous Databases,* (Venezia University, April 10-14, 1989).

43. Markowitz, V. M. & Raz, Y. A Modified Relational Algebra and Its Use in an Entity-Relationship Environment. In C. G. Davis, S. Jajodia, P. A. Ng, & R. T. Yeh, eds., *Entity-relationship approach to software engineering.* Elsevier Science Publishers, 1983, 315-328.

44. Markowitz, V. M. & Shoshani, A. Abbreviated query interpretation in extended entity-relationship oriented databases. In *Proceedings of the 8th International Conference* (Toronto, Canada, 1989) 40-58.

45. Markowitz, V. M. and Shoshani, A. On the Correctness of Representing Extended Entity-Relationship Structures in the Relational Model. *Proceedings of the 1989 SIDMOD Conference, SIDMOD Record 18,* 2, (June 1989), 430-439.

46. McLeod, D. J. High level domain definition in a relational data base system. *IBM Research Laboratory* (San Jose, CA., 1976).

47. Paget, M. L. A knowledge-based approach toward integrating international on-line databases. In *WP # CIS-89-01* (Sloan School of Management, MIT, Cambridge, MA. 1989). CISL Project.

48. Parent, C. & Spaccapietra, S. An Algebra for a General Entity-Relationship Model. *IEEE Transactions on Software Engineering, SE-11,* 7 (1985), 634-643.

49. Parent, C., Rolin, H., Yetongnon, K., Spaccapietra, S. An ER calculus for the entity-relationship complex model. In *Proceedings of the 8th International Conference* (Toronto, Canada, 1989). 75-98.

50. Peckham, J. & Maryanski, F. Semantic Data Models. *ACM Computing Surveys, 20* , 3 (1988), 153-189.

51. Porter, M. & Millar, V. E., How Information Gives You Competitive Advantages. *Harvard Business Review* (July-August, 1985), 149-160.

52. Qian, X. & Wiederhold, G. Knowledge-based integrity constraint validation. In *Proceedings of the Twelfth International Conference on Very Large Data Bases* (1986), 3-12.

53. Rockart, J. The Line Takes the Leadership: IS Management in a Wired Society. *Sloan Management Review, MIT, 29,* 4 (Spring 1988), 57-64.

54. Rusinkiewicz, M., Emasri, R., Czejdo, B., Georgakopoulos, D., Karabatis, G., Jamoussi, A., Loa, K., Li, Y., Gilbert, J., & Musgrove, R. Query processing in omnibase - a loosely coupled multi-database system. In the University of Houston *Technical Report #UH-CS-88-05.* (1988).

55. Shin, D. G. Semantics for handling queries with missing information. In *Proceedings of the Ninth International Conference on Information Systems 9* (1988), 161 - 167.

56. Smith, J. M., Bernstein, P. A., Dayal, U., Goodman, N., Landers, T., Lin, K.W.T., & Wong, E. *Multibase - Integrating Heterogeneous Distributed Database Systems.* 1981 National Computer Conference, 1981, 487-499.

57. Stonebraker, M. Inclusion of New Types in Relational Data Base Systems. *IEEE,* (1986),480-487.

58. Teorey, T. J., Yang, D., & Fry, J. P. A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model. *Computing Surveys, 18,* 2 (1986), 197-222.

59. Wang, Y. R. & Madnick, S. E. Evolution Towards Strategic Applications of Databases Through Composite Information Systems. *Journal of Management Information System* (Fall, 1988), 5-22.

60. Wang, Y. R. & Madnick, S. Connectivity Among information Systems. *Composite Information Systems (CIS) Project 1* (1988), 141 pages.

61. Wang, Y. R. & Madnick, S. Facilitating Connectivity in Composite Information Systems, To appear in *The ACM, Database* (1989a).

62. Wang, Y. R. & Madnick, S. The Inter-Database Instance Identification Problem in Integrating Autonomous Systems. In *Proceedings of the Fifth International Conference on Data Engineering,*(February 6-10, 1989.)

63. Weller D. L., & York, B. W. A Relational Representation of an Abstract Type System. *IEEE Transactions on Software Engineering SE-10,* 3 (May, 1984) 303-309.

64. Wong, T. K. Data connectivity for the Composite Information System/Tool Kit. *WP # CIS-89-03* (Sloan School of Management, MIT, Cambridge, MA. 1989), CISL Project.

65. Zaniolo, C. The Database Language GEM. *Readings in Object-Oriented Database Systems* (1990) 449-469.