A Methodology for Integration of Heterogeneous Databases

M.P. Reddy
B.E. Prasad
P.G. Reddy
Amar Gupta

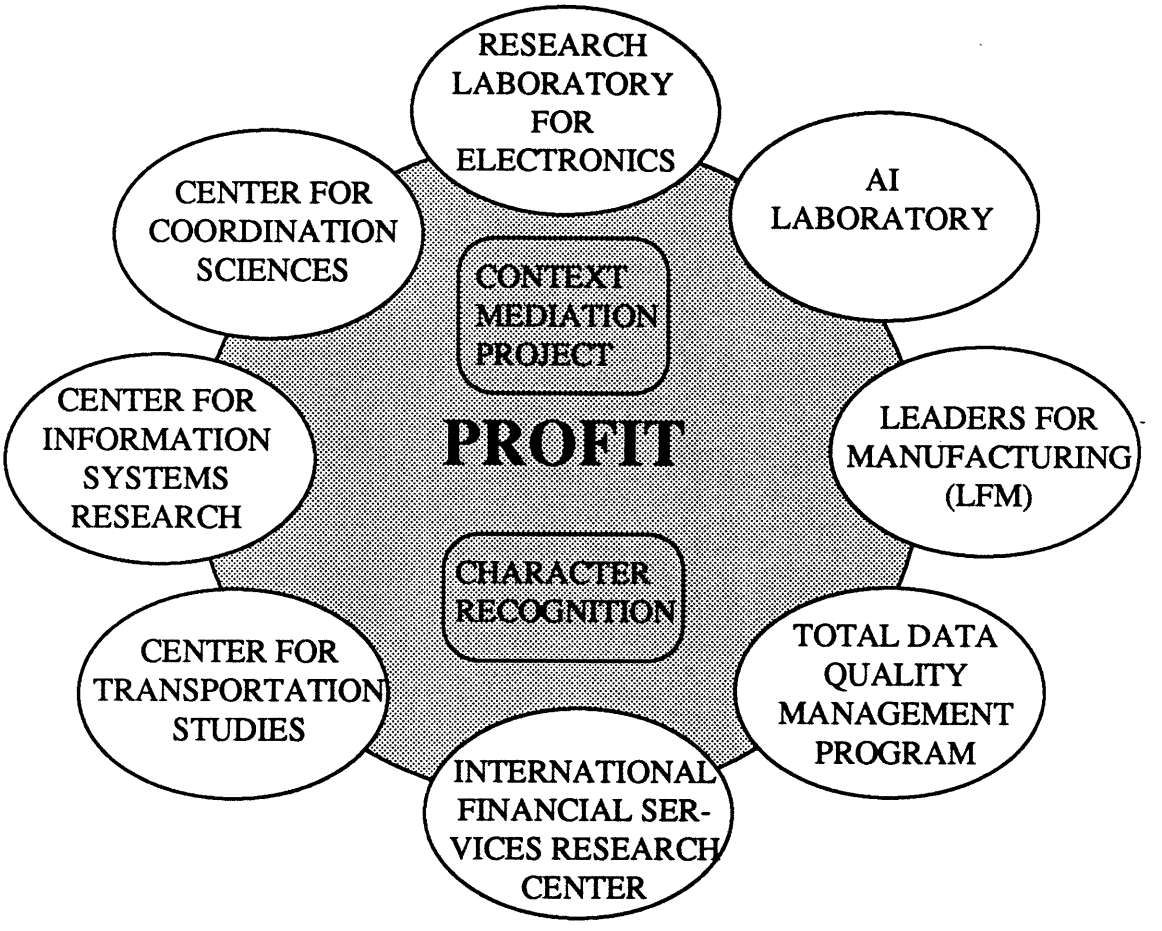
WP #3781 December 1994
PROFIT #94-20

Productivity From Information Technology
"PROFIT" Research Initiative
Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
(617)253-8584
Fax: (617)258-7579

 The research described herein has been supported (in whole or in part) by the Productivity From Information Technology (PROFIT) Research Initiative at MIT.

Productivity From Information Technology (PROFIT)

The Productivity From Information Technology (PROFIT) Initiative was established on October 23, 1992 by MIT President Charles Vest and Provost Mark Wrighton "to study the use of information technology in both the private and public sectors and to enhance productivity in areas ranging from finance to transportation, and from manufacturing to telecommunications." At the time of its inception, PROFIT took over the Composite Information Systems Laboratory and Handwritten Character Recognition Laboratory. These two laboratories are now involved in research related to context mediation and imaging respectively.



In addition, PROFIT has undertaken joint efforts with a number of research centers, laboratories, and programs at MIT, and the results of these efforts are documented in Discussion Papers published by PROFIT and/or the collaborating MIT entity.

Correspondence can be addressed to:

The "PROFIT" Initiative
Room E53-310, MIT
50 Memorial Drive
Cambridge, MA 02142-1247
Tel: (617) 253-8584
Fax: (617) 258-7579
E-Mail: profit@mit.edu

A Methodology for Integration of Heterogeneous Databases

M. P. Reddy, *Member, IEEE*, B. E. Prasad, *Member, IEEE*, P. G. Reddy, and Amar Gupta, *Senior Member, IEEE*

Abstract— **The transformation of existing local databases to meet diverse application needs at the global level is performed through a four-layered procedure that stresses total schema integration and virtual integration of local databases. The proposed methodology covers both schema integration and database integration, and uses a four-layered schema architecture (local schemata, local object schemata, global schema, and global view schemata) with each layer presenting an integrated view of the concepts that characterize the layer below. Mechanisms for accomplishing this objective are presented in theoretical terms, along with a running example. Object equivalence classes, property equivalence classes, and other related concepts are discussed in the context of logical integration of heterogeneous schemata, while object instance equivalence classes and property instance equivalence classes, and other related concepts are discussed for data integration purposes. The proposed methodology resolves naming conflicts, scaling conflicts, type conflicts, and level of abstraction, and other types of conflicts during schema integration, and data inconsistencies during data integration.**

Index Terms—**Heterogeneous databases, integrated databases, schema integration, semantic incompatibilities.**

I. Introduction

THE problem of integrating disparate information systems has been investigated by a number of researchers [1]–[17], and examples of prototypes include MULTIBASE [18], [19], MERMAID [20]–[22], ADDS [23], PRECI* [24]–[27], IMDAS [28], [29], NDS [30], MRDSM [31]–[34], IISS [35], and CALIDA [36]–[38]. The merits and demerits of different approaches were investigated in [39] and classified into three broad categories—total integration, partial integration, and interoperability. The subject of heterogeneous distributed database management systems (HDDBMS) has been discussed in [40]–[42].

This paper describes a systematic procedure for virtual integration of component databases; as compared to existing procedures, this procedure lays greater stress on resolving semantic incompatibilities among the component database schemata and data inconsistencies among data in the local databases. This methodology uses a four-layered schema architecture: local schemata, local object schemata, global schema, and global view schemata as shown in Fig. 1. Each layer presents an integrated view of the concepts that characterize the layer below.

The design methodology of each one of the four layers is discussed in the following four sections of the paper respectively. Section II focuses on local schemata and includes a discussion of incompatibilities that cause difficulty in integrating heterogeneous databases. Section III concentrates on local object schemata and formal notation needed for integration purposes. Section IV delineates one mechanism [43], [44] to integrate local object schemata to construct a global schema. Section V relates to the definition of views of the global schema. A procedure to construct instances of the objects in the global schema is presented in Section VI, while conclusions are presented in the last section of this paper.

II. Local Schema

The bottom layer consists of a set of local schemata. Each local schema can be denoted by D_i, where 'i' identifies the database. These schemata provide a description of the data stored in their respective data models. The stored data can be retrieved only by using their respective query languages. Structural and semantic incompatibilities occur in the component databases due to dissimilar perceptions of the same reality by different individuals, dissimilar data modeling skills of designers of various schemata, and differences in semantic richness of data models used in these schema definitions. In any decentralized environment, identification of common concepts is difficult because they frequently carry neither a common name nor a similar structure. The direct integration of a given set of local schema is difficult because of various kinds of semantic incompatibilities and data inconsistencies described in the following subsections.

A. Semantic Incompatibilities

- **Naming Conflicts:** In any data model, the schemata incorporate names for various entities/objects represented by them. Since these schemata are designed independently, the designer of each schema uses his or her own vocabulary to name these objects. Objects in different schemata representing the same real world concept may contain dissimilar names [1], [2], [14], [45], [7], [4] resulting in problems of two types:

Manuscript received February 24, 1993; revised July 28, 1993.

M. P. Reddy is with Kenan Systems, Cambridge, MA 02142 USA.
B. E. Prasad the Dow Jones Telerate, Jersey City, NJ 07311 USA.
P. G. Reddy is with the Department of Computer and Information Sciences, University of Hyderabad, Hyderabad, India.
A. Gupta is with the Sloan School of Management, Massachusettes Institute of Technology, Cambridge, MA 02139 USA.

IEEE Log Number 9213307.

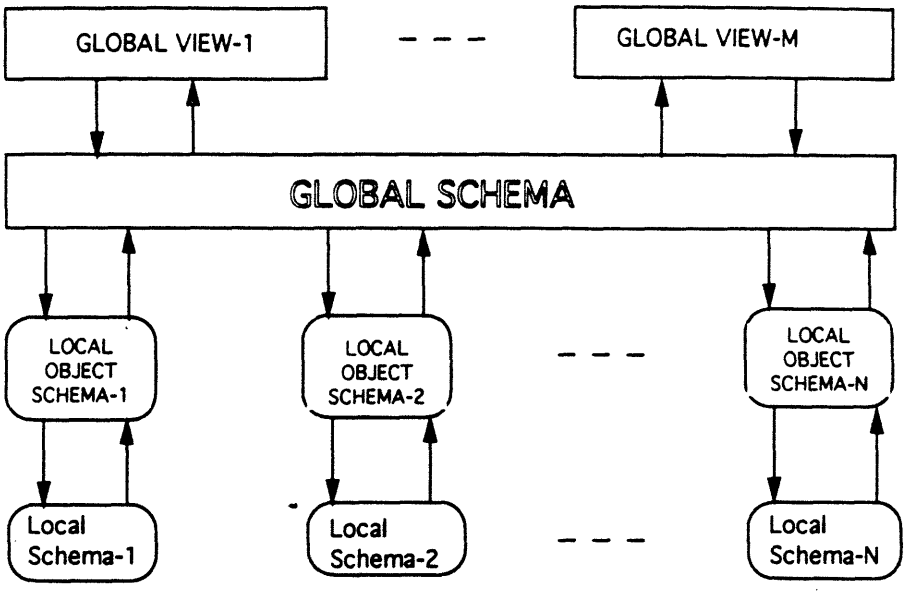


Fig. 1. Schema architectures of a four-layered HDDBMS.

— **Homonyms:** This inconsistency arises when the same name is used for two different concepts. For example, 'SALARY' may mean weekly salary in one database, and monthly salary in another.

— **Synonyms:** This type of naming conflict arises when the same concept is identified by two or more names. For example, the term 'DOMESTIC-CUSTOMER' in one database may refer to the same concept as the term 'BUYERS' in another database.

Note that homonyms and synonyms can only be detected by external specification.

- **Type Conflicts:** These conflicts arise when the same concept is represented by different coding constructs in different schemata [1]. For example, an object may be represented as an entity in one schema and as an attribute in another schema.
- **Key Conflicts:** Different keys may be assigned to the same concept in different schemata [15], [46]. For example, ss# and EMP-ID may be keys for employees in two component schemata.
- **Behavioral Conflicts:** These conflicts arise when different insertion/deletion policies are associated with the same class of objects in different schemata [15]. For example, in one database, the relation DEPT may exist without having any employee records being associated with it, where as in another database, the deletion of the last employee record may also delete the relation DEPT from the database.
- **Missing Data:** Different attributes may be defined for the same concept in different schemata [47]. For example, EMP1(SSN, NAME, AGE) and EMP2(SSN, NAME, ADDRESS) may represent the same concept in two database schemata. Attribute 'AGE' is missing in EMP2, and attribute 'ADDRESS' is missing in EMP1.
- **Levels of Abstraction:** This incompatibility is encountered when information about an entity is stored at dissimilar levels of detail in two databases [13]. For example, 'LABOR-COST' and 'MATERIAL-COST' may be stored separately in one database and combined together as 'TOTAL-COST' in a second database.
- **Identification of Related Concepts:** For concepts in the component schemata that are not the same but are related, one needs to discover all the inter-schema properties that relate to them. For example, two entities belonging to two different databases may not be equivalent but one entity may be a generalization of the other entity.
- **Scaling Conflicts:** This incompatibility arises when the same attribute of an entity is stored in dissimilar units in different databases [14]. For example, the attribute 'LENGTH' of an entity may be stored in terms of centimeters in one database and as inches in another database.

B. Quantitative Data Incompatibilities

Data retrieved from two local databases for the same logical data item may be incompatible for the following reasons:

- **Different Levels of Accuracy:** Different databases may be storing an attribute at dissimilar levels of accuracy. For example, one database may contain the weight of a particular part up to an accuracy of a milligram, whereas another database may specify accuracy only up to a gram [43].
- **Asyncronous Updates:** Since each database is managed independently, all databases may not update the value simultaneously [48].
- **Lack of Security:** Due to lack of information security at component databases, unauthorized users may have changed the data [49].

A local schema contains various relations that exist between the entities/concepts described in that schema. But the design

of an integrated schema requires relations among the entities of different database schemata. The types of semantic incompatibilities described in this section cause difficulty in establishing relations among the entities of different database schemata.

To mitigate these semantic incompatibilities, it is necessary to collect semantic knowledge regarding all the entities present in the component databases. The local schemata and the data-dictionaries alone cannot provide all the semantics of an entity because some of the semantics could be in default or implicit form. In order to overcome this problem, the global schema designer is required to collect the information missing in the local database's data-dictionary from the corresponding database administrator. This knowledge needs to be explicitly stored so that it can be utilized during the integration of these databases. The process of acquisition and representation of the semantic knowledge for a given schema is described in the next section.

III. Modeling of Local Object Schema

In this section, local schemata are analyzed and parameters useful for integration are derived. All these parameters are explicitly stored in the form of *local object schema.*

There are three motivations for designing a local object schema: first, to achieve data model homogeneity among the component database schemata [50]; second, to present clear and logical views of entities present in the local schemata; and third, to store all the semantic knowledge gathered about the object in the local schema. These issues are elaborated in the following paragraphs.

To facilitate the identification of equivalent entities in different database schemata, the schemata must first be translated into a single data model for comparison purposes. The primary objectives for constructing local object schema are to achieve data model homogeneity by transforming local schemata expressed in different data models into a single data model and to achieve uniformity in the modeling constructs. The construction of a global schema is easier if all the local schemata pertain to the same data model.

The secondary goal is to present a clear logical view of the entities in the local schema. In a local schema, information pertinent to an entity may be scattered over more than one data structure[1] for various reasons. In order to obtain a clear perception of the entity, one wishes to view all the information available for that entity as a single unit (i.e., to have an integrated view of the entity irrespective of its distribution in the local schema).

A local object schema is also meant to explicitly represent all underlying assumptions in the corresponding local schema. For example, each user of a particular database may be aware that the value returned by the property EMP-SAL of an entity EMP is the monthly salary paid in dollars, even though the currency is not explicitly represented in the database schema. Assumptions underlying a particular local schema can be explicitly stored in the corresponding local object schema.

As such, for any given local component database schema D_i, its corresponding local object schema OD_i needs to be constructed. The creation of OD_i involves the derivation of four parameters, namely, S_i, ψ^i, SK_i and MK_i from the corresponding local schema. The set S_i contains distinct object types together with their properties in OD_i and the matrix ψ^i specifies relations among the object types in S_i. The parameter SK_i denotes the *semantic knowledge* gathered for the local object schema. MK_i incorporates the *mapping knowledge* that is necessary to construct instances of the objects in S_i from the data stored under D_i. A local object schema OD_i can therefore be written as a quadruple, $OD_i = \{S_i, \psi_i, SK_i, MK_i\}$. These concepts are explained further in the following subsections.

A. Object Types in Local Object Schema (S_i)

An object in a local object schema is any distinguishable entity whose description is available in the local schema D_i. In other words, all the information pertaining to an entity which is scattered among various data structures in the local database schema is represented by a single object in the local object schema.

A database object is denoted by O_l, where l is a unique object identifier; l is a pair, say (i.o), where the first index 'i' gives the schema identification and second index 'o' gives the object identification within the schema. For example, if the object EMP is denoted by $O_{1.1}$, it means that the object EMP belongs to the database schema OD_1, and its identification in that schema is '1.'

The first parameter S_i in the definition of local object schema OD_i is defined as all distinct object types whose description is available in the local database schema D_i. For example, if the database schema D_1 provides the description of three objects namely, EMP, FACULTY, DEPT, then S_1 can be written as

$$S_1 = \{\text{EMP, FACULTY, DEPT}\} = \{O_{1.1}, O_{1.2}, O_{1.3}\}$$

Similarly, objects in D_2 and D_3 can be written as follows:

$$S_2 = \{\text{EMPLOYEE, VISITING-PROF, PROFESSOR}\} = \{O_{2.1}, O_{2.2}, O_{2.3}\}$$

$$S_3 = \{\text{TEMPORARY-STAFF, V-PROF, DIVISION}\} = \{O_{3.1}, O_{3.2}, O_{3.3}\}.$$

Each object has a set of properties associated with it. A property of an object is denoted by P_k where 'k' is a unique property identifier; k is a pair $l.pi$ where l is its object identifier and pi is the property identifier with respect to the object O_l. The property set associated with the object O_l is denoted by PS_{O_l}. The object O_l is characterized by its properties. This characterization is denoted by $O_l \Leftrightarrow PS_{O_l}$. For example, one can write

[1] In a relational model sense, over more than one relation.

$$O_{1.1} \Leftrightarrow \{\text{EMP-ID, EMP-SAL, DEPT-NAME}\}$$
$$\Leftrightarrow \{P_{1.1.1}, P_{1.1.2}, P_{1.1.3}\}.$$

This provides the method for the derivation of the S_i component of the local object schema definition. The procedure for representing relations among the objects in S_i is presented in the next subsection.

B. Relations Among Objects of a Local Object Schema (ψ_i)

Various relations among the objects of S_i are captured through the matrix ψ_i. This matrix is intended to be generated by the local database administrator based solely on his/her local database without knowledge of any details about the other database schemata that are planned to be integrated during the definition of the relation matrix.

Given two objects O_l and O_k, the relationship among the sets of their instances can be classified into the following:

Category-I: equivalence relation.
Category-II: super class-subclass relation.
Category-III: overlapping relation.
Category-IV: disjoint relation.

In a single database schema, object equivalence cannot exist since a database schema represents a non-redundant view of the object types. As such, only relations in other categories need to be captured.

Each element in the matrix ψ^i is a tuple $\langle \sigma, \delta \rangle$. Given $O_l, O_k \in S_i$, the relationship among these two objects is given by $\psi_{l,k}$ of the matrix ψ^i. An element $\psi_{l,k}$ of matrix ψ^i is defined as follows:

- Subclass-Super class Relation

$$\psi_{l,k} \cdot \sigma = 1 \text{ if } O_l \text{ has a subclass relation with } O_k$$
$$= 0 \text{ otherwise.}$$

In the example, consider the object schema OD_1. Each faculty member is an employee. As such, FACULTY has a subclass relation with EMP, which is represented as $\psi_{\text{FACULTY,EMP}} \cdot \sigma = 1$.

- Disjoint or Overlap Relation

$$\psi_{l,k} \cdot \delta = 1 \text{ if } O_l \text{ is disjoint with } O_k.$$
$$= 0 \text{ if } O_l \text{ overlaps with } O_k.$$

For example, EMP and DEPT are disjoint, that is, $\psi_{\text{EMP,DEPT}} \cdot \delta = 1$.

The relation matrix ψ^i for the local object schema OD_i is constructed as follows:

$$\psi^i = \bigcup_{O_l, O_k \in S_i} \psi_{lk}.$$

The relationship matrix ψ^1 constructed from the database schema OD_1 is shown below.

$$\psi^1 = \begin{array}{c} \\ O_{1.1} \\ O_{1.2} \\ O_{1.3} \end{array} \begin{array}{c} \begin{array}{ccc} O_{1.1} & O_{1.2} & O_{1.3} \end{array} \\ \begin{pmatrix} \langle 1,0\rangle & \langle 0,0\rangle & \langle 0,1\rangle \\ \langle 1,0\rangle & \langle 1,0\rangle & \langle 0,1\rangle \\ \langle 0,1\rangle & \langle 0,1\rangle & \langle 1,0\rangle. \end{pmatrix} \end{array}$$

Similarly, ψ^2 and ψ^3 for database schemata OD_2 and OD_3 are shown below.

$$\psi^2 = \begin{array}{c} \\ O_{2.1} \\ O_{2.2} \\ O_{2.3} \end{array} \begin{array}{c} \begin{array}{ccc} O_{2.1} & O_{2.2} & O_{2.3} \end{array} \\ \begin{pmatrix} \langle 1,0\rangle & \langle 0,0\rangle & \langle 0,0\rangle \\ \langle 1,0\rangle & \langle 1,0\rangle & \langle 1,0\rangle \\ \langle 1,0\rangle & \langle 0,0\rangle & \langle 1,0\rangle \end{pmatrix} \end{array}$$

$$\psi^3 = \begin{array}{c} \\ O_{3.1} \\ O_{3.2} \\ O_{3.3} \end{array} \begin{array}{c} \begin{array}{ccc} O_{3.1} & O_{3.2} & O_{3.3} \end{array} \\ \begin{pmatrix} \langle 1,0\rangle & \langle 0,0\rangle & \langle 0,0\rangle \\ \langle 1,0\rangle & \langle 1,0\rangle & \langle 0,1\rangle \\ \langle 0,1\rangle & \langle 0,1\rangle & \langle 1,0\rangle. \end{pmatrix} \end{array}$$

In this manner ψ^i can be constructed for any local object schema definition.

C. Semantic Knowledge About Objects in Local Object Schema SK_i

The third parameter SK_i requires semantic knowledge pertinent to each property of each object belonging to OD_i. This semantic knowledge relating to each property is captured through its set of *meta-properties* and their corresponding *meta-values* defined in the following paragraphs.

Meta-properties are the parameters necessary to provide a semantic meaning (without any ambiguity) to the symbols/values associated with the property of an oject. For example, PERIODICITY-OF-PAY and CURRENCY are some of the meta-properties of the property EMP-SAL.

Let M_k^i denote the 'i'th meta-property of the property P_k. Let M^{P_k} denote the set of meta-properties associated with the property P_k and $|M^{P_k}|$ denote the number of meta-properties associated with P_k. For each meta-property, there is a set of legal values associated with it. For example, DOLLAR, RUPEE, and POUND are some of the legal values of the property CURRENCY; similarly WEEKLY, MONTHLY, and YEARLY are some of the legal values of the meta-property PERIODICITY-OF-PAY.

Define $M_k^i(P_k) = V_k^i$ if V_k^i is the meta-value of the property P_k associated with the meta-property M_k^i. For any value of k, the *semantic equation* of the property P_k denoted by SE_{P_k} is defined as follows:

$$SE_{P_k} = \left\{ \{\text{INST}_i(P_k)\}_i, \left\{ (M_k^1, V_k^1), (M_k^2, V_k^2), \cdots \left(M_k^{|M^{P_k}|}, V_k^{|M^{P_k}|}\right), C_k \right\} \right\}$$

where $\{\text{INST}_i(P_k)\}_i$ is the set of values explicitly stored in the database as data, $V_k^1, V_k^2, \cdots, V_k^{|M^{P_k}|}$ are the meta values of the meta properties $M_k^1, M_k^2, \cdots, M_K^{|M^{P_k}|}$ respectively, and C_k represents the constraints pertinent to the property P_k such as its range, legal values, and deletion/insertion permissions. These values are implicitly stored in the database as meta-data.

We define the semantic equation for an object O_l, which is denoted by SE_{O_l} as follows:

$$SE_{O_l} = \{SE_{P_k} | P_k \in PS_{O_l}\}.$$

The above equation can be rearranged and written as

$$SE_{O_l} = \left\{ \{\text{INST}_i(O_l)\}_i, \left\{ \left\{ (M_k^1, V_k^1), (M_k^2, V_k^2), \cdots, \left(M_k^{|M^{P_k}|}, V_k^{|M^{P_k}|}\right) \right\}, C_k \right\}_k \right\}$$

where $\text{INST}_i(O_l) = \{\text{INST}_i(P_k) | P_k \in \text{PS}_{O_l}\}$ and further $\text{INST}_i(P_k)$ can be derived from $\text{INST}_i(O_l)$ as follows:

$$\text{INST}_i(P_k) = P_k(\text{INST}_i(O_l)).$$

The first parameter in the eqution for SE_{O_l} represents the aggregation of all its instances, and the second parameter gives the aggregation of meta-data of all its properties. The semantic knowledge SK_i of any local object schema can be derived using the equation:

$$SK_i = \{SE_{O_l} | O_l \in S_i\}$$

Finally, the last parameter in the definition of a local object schema can be derived as described in the following subsection.

D. Mapping Knowledge (MK_i)

Let MK_{O_l} represent the mapping knowledge, which provides details about how O_l is mapped to the data structures in the local schema. A specific procedure for the derivation of MK_{O_l} is not presented in this paper since it varies for each local data model. One procedure is described in [50]. The mapping knowledge MK_i is defined as

$$MK_i = \{MK_{O_l} | O_l \in S_i\}.$$

Note that the local object schema can be generated independently without considering any idiosyncrasies of the remaining component database schemata of the global schema. From a given set of local schemata, a global schema can be constructed as described in the next section.

IV. Construction of Global Schema

The derivation of the global schema, D_G, involves the derivation of four parameters, namely, S_G, ψ^G, SK_G and MK_G from their corresponding parameters in the local object schemata considered for integration. We describe the process for their derivations in the following subsections. In this paper, we denote objects and their properties in the global schema with bold type face letters, whereas objects and properties of a local object schema are denoted in normal type.

A. Object Types in Global Schema

Let U be the set of local object schemata considered for integration. The complete set of object types, denoted by CSO, is given by

$$\text{CSO} = \bigcup_{OD_d \in U} S_d$$

For the database schemata OD_1, OD_2, and OD_3, the complete set of objects is given by

$$\begin{aligned}\text{CSO} &= \{\text{EMP, FACULTY, DEPT, EMPLOYEE, VISITING-PROF, PROFESSOR, TEMPORARY-STAFF, V-PROF, DIVISION}\} \\ &= \{O_{1.1}, O_{1.2}, O_{1.3}, O_{2.1}, O_{2.2}, O_{2.3}, O_{3.1}, O_{3.2}, O_{3.3}\}.\end{aligned}$$

The set of object types in the global schema S_G is constructed from the objects of CSO. As each S_d represents a set of objects from an independent database schema, there exists the possibility of having object type redundancy in CSO. As D_G is also supposed to provide a non-redundant view of the world of discourse, S_G is expected to contain non-redundant object types. Hence, all redundant object types in CSO need to be identified.

To identify object type redundancy, two object types cannot be compared directly, because each schema is usually designed and managed independently of others, causing various kinds of heterogeneity and data incompleteness problems. Given two objects belonging to two different schemata, neither their names nor their structures can be compared directly to determine whether they are equivalent or not. The concept of object-identity [51] has been proposed to establish equivalence among different versions of the same object in the context of object-oriented databases; however, such techniques are not useful since our attempt is to integrate data belonging to different databases, mostly from traditional databases, where the concept of object identity is not defined.

The notion of real world states (RWS's) [52], [6] of objects, that is, the scope of the real world they are designed to reflect in the database, is used to compare two objects. When two objects A and B in two different databases actually represent the same sets of instances of the same real world entity, they are deemed to possess the same real world states i.e., $\text{RWS}(A) = \text{RWS}(B)$. However, instances of A need not provide the same information as the corresponding instances of B. For example, two objects, EMP(EMP-ID, EMP-SAL, EMP-DEPT) and EMPLOYEE (E-ID, E-SAL, DIV-NAME, E-RANK), belonging to two different databases and providing information about the same set of employees contain dissimilar properties. Even when the property sets are the same, some instances may be missing for one of the objects. This makes it difficult to mechanize the identification of equivalence among objects in different schemata. Tools [53], [54], [8] developed to aid the designer in judging object equivalence can suggest the

possibility of equivalence, but only the designer can establish the equivalences by introducing external knowledge about the world of discourse.

Using the above comparison criterion, an equivalence relation '$\equiv$' on CSO is defined as follows:

For any two objects O_l, $O_k \in$ CSO, $O_l \equiv O_k$ if and only if RWS(O_l) = RWS(O_k). Note that the equivalence relation can be established among O_l and O_k by the global schema designer utilizing external knowledge about the world of discourse even when O_l and O_k have different property sets or have different sets of instances associated with them.

Suppose object equivalence assertions made on the objects of CSO are as follows:

RWS(EMP) = RWS(EMPLOYEE),
RWS(FACULTY) = RWS(FACULTY),
RWS(DEPT) = (DIVISION),
RWS(VISITING-PROF) = RWS(V-PROF), and
RWS(TEMPORARY-STAFF) = RWS(TEMPORARY-STAFF).

Using these relationships, one can decompose the set CSO into disjoint equivalence classes. The resultant set is denoted by CSO_E. For any O_l belonging to CSO, the equivalence class to which O_l belongs is denoted by $[O_l]$.

$$[O_l] = \{O_k | O_k \in \text{ CSO and } O_k \equiv O_l\}$$
$$CSO_E = \{[O_l] | O_l \in \text{CSO}\}.$$

Accordingly, the following equivalence classes apply:

[EMP] = {EMP, EMPLOYEE},
[FACULTY] = {FACULTY},
[DEPT] = {DEPT, DIVISION},
[PROFESSOR] = {PROFESSOR},
[VISITING-PROF] = {VISITING-PROF, V-PROF}, and
[TEMPORARY-STAFF] = {TEMPORARY-STAFF}.

As mentioned earlier, each object equivalence class in CSO_E provides the set of equivalent object types in CSO. As such, all objects in one object equivalence class need to be integrated to obtain a single global object type. Hence, one global object type $\mathbf{O_L}$ needs to be constructed from each object equivalence class $[O_l]$ belonging to CSO_E. The number of object types in S_G is therefore equal to the number of equivalence classes present in CSO_E. In this manner, one can obtain the set S_G with non-redundant object types.

If **EMPLOYEE, FACULTY, DEPT, PROFESSOR, VISITING-PROFESSOR,** and **TEMPORARY-STAFF** are the global objects derived from the object equivalence classes [EMP], [FACULTY], [DEPT], [PROFESSOR], [VISITING-PROF], and [TEMPORARY-STAFF] respectively, the global object types are as follows:

$$S_G = \{\textbf{EMPLOYEE, FACULTY, DEPT, PROFESSOR, VISITING-PROFESSOR, TEMPORARY-STAFF}\} = \{\mathbf{O_1}, \mathbf{O_2}, \mathbf{O_3}, \mathbf{O_4}, \mathbf{O_5}, \mathbf{O_6}\}.$$

The local object $O_k \in [O_l]$ from which the global object $\mathbf{O_L}$ is constructed is said to be a component of $\mathbf{O_L}$. This relation is symbolically denoted by $O_k \overset{*}{\in} \mathbf{O_L}$. Note that the real world state of $\mathbf{O_L}$ is that of any (hence all) $O_k \overset{*}{\in} \mathbf{O_L}$.

In our example, since EMP and EMPLOYEE are components of **EMPLOYEE**,

$$\text{RWS(EMPLOYEE)} = \text{RWS(EMP)} = \text{RWS(EMPLOYEE)}.$$

While all objects in S_G can be derived directly, the derivation of properties of a global object requires knowledge about the relationships among global objects. A mechanism to derive relationships among global objects is presented in the next subsection, and the procedure to derive properties of a global object is presented in Section IV-C.

B. Derivation of Relations Among Objects in Global Schema

In our endeavor to create the integrated database schema D_G, the set of global object types S_G was constructed. One now needs to determine the relationships among these objects, that is, the relationship matrix ψ^G needs to be constructed. The matrix ψ^G is constructed partially in an automatic way from the existing set of relationship matrices and partially through interaction with the designer as described in the following subsections.

1) Automatic Derivation of Relations among Global Object Types: For automatic derivation of relations among the global object types, the sum of two ψ's is defined as follows:

$$\psi_{l,k} + \psi_{m,n} = \langle \psi_{l,k} \cdot \sigma + \psi_{m,n} \cdot \sigma, \psi_{l,k} \cdot \delta + \psi_{m,n} \cdot \delta \rangle.$$

If O_l and O_k belong to the same schema, say OD_i, then the values of $\psi_{lk} \cdot \sigma$ and $\psi_{lk} \cdot \delta$ can be obtained from the relationship matrix ψ^i. If O_l and O_k belong to different schemata, then the values of $\psi_{l,k} \cdot \sigma$ and $\psi_{l,k} \cdot \delta$ are set to ∞ (unknown). The summation rules used while constructing ψ^G for the global objects are given by the matrix S. An element $S(\psi_{l,k} \cdot \sigma, \psi_{m,n} \cdot \sigma)$ in this matrix gives the value of $\psi_{l,k} \cdot \sigma + \psi_{m,n} \cdot \sigma$.

$$S = \begin{array}{c} \\ 1 \\ 0 \\ \infty \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 0 & \infty \end{array} \\ \begin{pmatrix} 1 & \text{Conflict} & 1 \\ \text{Conflict} & 0 & 0 \\ 1 & 0 & \infty \end{pmatrix} \end{array}$$

The result of the summation is a 'Conflict' in two cases; either one of the local object schema is inconsistent or the inter-schema object equivalence classes contain a fault. As such these two aspects will need to be investigated whenever such conflicts arise. The same summation rules apply for computing $\psi_{l,k} \cdot \delta + \psi_{m,n} \cdot \delta$.

Since the relationships between the objects of S_G are unknown initially, set $\psi^G_{\mathbf{L},\mathbf{K}} = \langle \infty, \infty \rangle$ for all $\mathbf{O_L}, \mathbf{O_K} \in S_G$.

The relationships between the objects of S_G are derived using the relationship matrices of the local schemata.

$$\psi^G_{\mathbf{L},\mathbf{K}} = \psi^G_{\mathbf{L},\mathbf{K}} + \sum_{O_l \overset{*}{\in} \mathbf{O_L}, O_k \overset{*}{\in} \mathbf{O_K}} \psi_{l,k}.$$

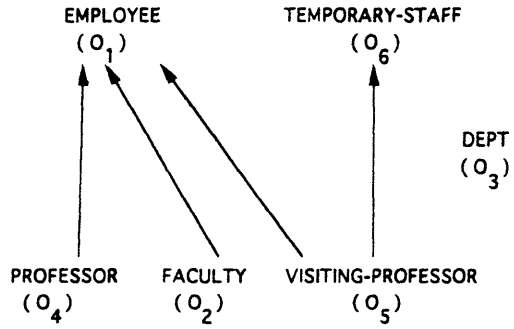


Fig. 2. Automatically derived subclass relations among global objects.

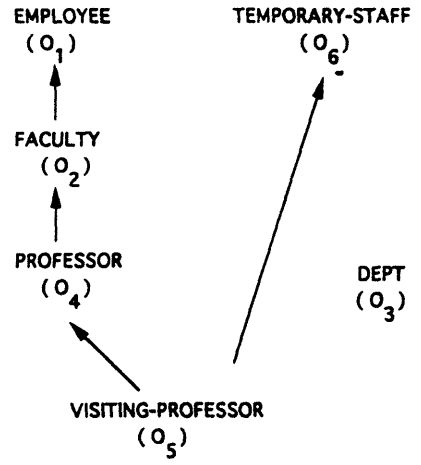


Fig. 3. Subclass relations among global objects.

The relationship matrix ψ^G for the global objects derived in the previous section from the local object schemata OD_1, OD_2, and OD_3 is shown at the bottom of the page. Subclass relations among the global object types derived during the above process are shown in Fig. 2. The arrow from **PROFESSOR** to **EMPLOYEE** represents the fact that the former object is a subclass of the latter object. For simplicity, overlap and disjoint relations among the objects are not shown in this figure.

The relationship matrix ψ^G constructed using the above suggested procedure cannot provide all the relations existing among the global object types. If two global objects $\mathbf{O_L}$ and $\mathbf{O_K}$ do not have any of their components together in a single database schema, then the relations between $\mathbf{O_L}$ and $\mathbf{O_K}$ cannot be determined from ψ^d's. Such missing relations in ψ^G need to be established. A procedure to establish such missing relations is described in the next subsubsection.

2) Interactive Specification of Missing Relationships in ψ^G: Relationships between the objects of S_G which cannot be derived through the automatic process need to be established through interaction with the designer.

If $\psi^G_{\mathbf{L},\mathbf{K}} = \langle \infty, \infty \rangle$, then the relationships between the objects $\mathbf{O_L}$ and $\mathbf{O_K}$ are unknown. The designer can establish these relations by comparing their real world states. For example, the designer may specify that the object **PROFESSOR** has a subclass relation with the object **FACULTY**; as such $\psi_{4,2} = \langle 1, 0 \rangle$. The complete relationship matrix ψ^G for the global objects derived in our example is shown in the following matrix:

$$\psi^G = \begin{array}{c} \\ \mathbf{O_1} \\ \mathbf{O_2} \\ \mathbf{O_3} \\ \mathbf{O_4} \\ \mathbf{O_5} \\ \mathbf{O_6} \end{array} \begin{array}{c} \begin{array}{cccccc} \mathbf{O_1} & \mathbf{O_2} & \mathbf{O_3} & \mathbf{O_4} & \mathbf{O_5} & \mathbf{O_6} \end{array} \\ \begin{pmatrix} \langle 1,0\rangle & \langle 0,0\rangle & \langle 0,1\rangle & \langle 0,0\rangle & \langle 0,0\rangle & \langle 0,0\rangle \\ \langle 1,0\rangle & \langle 1,0\rangle & \langle 0,1\rangle & \langle 0,0\rangle & \langle 0,0\rangle & \langle 0,0\rangle \\ \langle 0,1\rangle & \langle 0,1\rangle & \langle 1,0\rangle & \langle 0,1\rangle & \langle 0,1\rangle & \langle 0,1\rangle \\ \langle 1,0\rangle & \langle 1,0\rangle & \langle 0,1\rangle & \langle 1,0\rangle & \langle 0,0\rangle & \langle 0,0\rangle \\ \langle 1,0\rangle & \langle 1,0\rangle & \langle 0,1\rangle & \langle 1,0\rangle & \langle 1,0\rangle & \langle 1,0\rangle \\ \langle 1,0\rangle & \langle 0,0\rangle & \langle 0,1\rangle & \langle 0,0\rangle & \langle 0,0\rangle & \langle 1,0\rangle \end{pmatrix} \end{array}$$

The above relationships are shown in Fig. 3. In general, there may be some inconsistencies among the relations derived among global objects. The identification of such inconsistencies is discussed in the next subsubsection.

3) Checking Consistency of Global Schema: The derived relationships are subjected to consistency checks discussed in [55] and the tests are as follows:

1) If $\psi^G_{\mathbf{L},\mathbf{K}} \cdot \sigma = 1$ and $\psi^G_{\mathbf{L},\mathbf{K}} \cdot \delta = 1$ or $\psi^G_{\mathbf{K},\mathbf{L}} \cdot \delta = 1$, (i.e., $\mathbf{O_L}$ is a subclass of $\mathbf{O_K}$ and $\mathbf{O_L}$ is disjoint with $\mathbf{O_K}$), then the relation $\psi^G_{\mathbf{L},\mathbf{K}}$ is inconsistent. If $\mathbf{O_L}$ is a subclass of $\mathbf{O_K}$, then RWS($\mathbf{O_L}$) $\subset$ RWS($\mathbf{O_K}$) whereas if $\mathbf{O_L}$ is disjoint with $\mathbf{O_K}$, then RWS($\mathbf{O_L}$) $\cap$ RWS($\mathbf{O_K}$) $= \phi$, and these two conditions cannot be satisfied simultaneously. As such, a schema having such relations is inconsistent.
2) If $\psi^G_{\mathbf{L_1},\mathbf{L_2}} \cdot \sigma = \psi^G_{\mathbf{L_2},\mathbf{L_3}} \cdot \sigma = \cdots = \psi^G_{\mathbf{L_n},\mathbf{L_{n+1}}} \cdot \sigma = \psi^G_{\mathbf{L_{n+1}},\mathbf{L_1}} \cdot \sigma = 1$, then the schema is inconsistent. Cycles in subclass relations should not exist. If such cycles exist, then $\mathbf{O_{L_1}}, \mathbf{O_{L_2}}, \cdots, \mathbf{O_{L_{n+1}}}$ are all identical and should be merged into a single object.

The above inconsistency may arise because of two reasons: there may be a mistake in inter-schema object equivalence assertions or one of the component schema may be inconsistent. All inconsistencies must be analyzed either by looking at relations between local schema objects or by looking at previous equivalence assertions. The relations among global

$$\psi^G = \begin{array}{c} \\ \mathbf{O_1} \\ \mathbf{O_2} \\ \mathbf{O_3} \\ \mathbf{O_4} \\ \mathbf{O_5} \\ \mathbf{O_6} \end{array} \begin{array}{c} \begin{array}{cccccc} \mathbf{O_1} & \mathbf{O_2} & \mathbf{O_3} & \mathbf{O_4} & \mathbf{O_5} & \mathbf{O_6} \end{array} \\ \begin{pmatrix} \langle 1,0\rangle & \langle 0,0\rangle & \langle 0,1\rangle & \langle 0,0\rangle & \langle 0,0\rangle & \langle \infty,\infty\rangle \\ \langle 1,0\rangle & \langle 1,0\rangle & \langle 0,1\rangle & \langle \infty,\infty\rangle & \langle \infty,\infty\rangle & \langle \infty,\infty\rangle \\ \langle 0,1\rangle & \langle 0,1\rangle & \langle 1,0\rangle & \langle \infty,\infty\rangle & \langle 0,1\rangle & \langle 0,1\rangle \\ \langle 1,0\rangle & \langle \infty,\infty\rangle & \langle \infty,\infty\rangle & \langle 1,0\rangle & \langle 0,0\rangle & \langle \infty,\infty\rangle \\ \langle 1,0\rangle & \langle \infty,\infty\rangle & \langle 0,1\rangle & \langle \infty,\infty\rangle & \langle 1,0\rangle & \langle 1,0\rangle \\ \langle \infty,\infty\rangle & \langle \infty,\infty\rangle & \langle 0,1\rangle & \langle \infty,\infty\rangle & \langle 0,0\rangle & \langle 1,0\rangle \end{pmatrix} \end{array}$$

objects derived in this subsection are used to derive properties of a global object. A mechanism to derive properties of a global object is presented in the next subsection.

C. Derivation of Properties of Objects in Global Schema

In Section IV-A, object types in the global schema were identified, but properties of these objects were not derived. In this subsection, a procedure to derive the properties of an object is presented, after defining relevant terms.

•Property Equivalence: Determining if two properties of two objects are equivalent is as difficult as determining if two objects are equivalent. Two properties are defined to be equivalent if they describe the same characteristic of the corresponding real world concept. If P_k is equivalent to $P_{k'}$, then it is denoted by $P_k \approx P_{k'}$. The set of all properties equivalent to $P_{k'}$ is denoted by $[P_{k'}]$. Considering the properties of two objects EMP and FACULTY, and knowing that EMP-SAL is equivalent to FAC-PAY, we can denote this fact by EMP-SAL $\approx$ FAC-PAY. As such [EMP-SAL] = {EMP-SAL, FAC-PAY}.

In general, one can consider equivalence among the set of equivalence classes as well. If $P_x \in [P_x]$ and $P_y \in [P_y]$ are equivalent as properties, then $[P_x]$ is equivalent to $[P_y]$ and is expressed by $[P_x] \approx [P_y]$. Note that $[P_x] \cup [P_y]$ is an equivalence class. Occasionally, a set of 'n' properties $P_{k_1}, \cdots, P_{k_n}$ of an object is equivalent to set of 'm' properties $P_{l_1}, \cdots, P_{l_m}$ of another object, and this fact can be denoted by $\{P_{k_1}, \cdots, P_{k_n}\} \approx \{P_{l_1}, \cdots, P_{l_m}\}$. This type of incompatibility is caused by dissimilar abstractions. For example, the property EMP-SAL of the object EMP is equivalent to the combination of the two properties WORKING-HOURS and WAGES of the object TEMPORARY-STAFF.

One notices from the above definition of property equivalence that a property of an object can be equivalent to the set of all properties of another object put together. In other words, a single property of an object can be equivalent to another object, which means that a particular concept has been modelled as a property in one schema and as an object in another schema. This type of incompatibility is termed type incompatibility, and can be handled with this model.

With the above definition of property equivalence, properties of an object type can be classified into two categories: own properties versus acquired/inherited properties. The properties of $\mathbf{O_L}$ that are derived from its component objects are called own-properties of $\mathbf{O_L}$. The global object $\mathbf{O_L}$ is partially constructed by integrating its components. In other words, its own properties are constructed by integrating the properties of all its components.

Apart from its own properties, each object acquires a set of properties from objects having a superclass relation with the object through inheritance. Such properties are called acquired/inherited properties.

1) Own Properties of Global Object Type: Let O_l and $O_{l'}$ be two objects. Let PS_{O_l} and $PS_{O_{l'}}$ be the set of properties associated with objects O_l and $O_{l'}$ respectively. The operator $\dot{\cup}$ on PS_{O_l} and $PS_{O_{l'}}$ pertinent to global property derivation can be defined as follows:

$$[P_x] = \{P_y | P_y \in (PS_{O_l} \cup PS_{O_{l'}}) \text{ such that } P_x \approx P_y\}$$

$$PS_{O_l} \dot{\cup} PS_{O_{l'}} = \{[P_x] | P_x \in (PS_{O_l} \cup PS_{O_{l'}}\}.$$

From the above definition, one can see that $\dot{\cup}$ is a binary operator which computes the union of its operandi (two property sets) and decomposes this union into a set of property equivalence classes. The application $\dot{\cup}$ on two property sets PS_{EMP} = {EMP-ID, EMP-SAL, DEPT-NAME} and PS_{EMPLOYEE} = {E-ID, E-SAL, DIV-NAME, E-RANK} will result in

$$PS_{\text{EMP}} \dot{\cup} PS_{\text{EMPLOYEE}} = \{\{\text{EMP-ID,E-ID}\}, \{\text{EMP-SAL, E-SAL}\}, \{\text{E-RANK}\}, \{\text{DEPT-NAME, DIV-NAME}\}\}.$$

Let $\text{OPS}_{\mathbf{O_L}}$ be the own property set of the object $\mathbf{O_L}$. Here, $\mathbf{O_L}$ is characterized with its own properties (properties derived from its own component objects), and let

$$\overline{\text{OPS}_{\mathbf{O_L}}} = \dot{\bigcup}_{O_l \in \mathbf{O_L}} PS_{O_l}$$

Then $\overline{\text{OPS}_{\text{EMPLOYEE}}}$ = {{EMP-ID, E-ID}, {E-SAL}, {E-RANK}, {DEPT-NAME, DIV-NAME}.}

From each property equivalence class in $\overline{\text{OPS}_{\mathbf{O_L}}}$, one property for the global object $\mathbf{O_L}$ can be derived. Let $\mathbf{P_X}$ be the global property derived from the property equivalence class $[P_x]$. This relationship is denoted by $\mathbf{P_X} \approx [P_x]$. Assume that the global properties derived from respective equivalence classes are as follows:

EMP-ID $\approx$ {EMP-ID, E-ID},
EMP-SAL $\approx$ {EMP-SAL, E-SAL},
DEPT-NAME $\approx$ {DEPT-NAME, DIV-NAME},
EMP-RANK $\approx$ {E-RANK}.

The set of own properties of a global object is given by the expression

$$\text{OPS}_{\mathbf{O_L}} = \{\mathbf{P_X} | \mathbf{P_X} \approx [P_x] \text{ and } [P_x] \in \overline{\text{OPS}_{\mathbf{O_L}}}\}$$

and own properties of global object **EMPLOYEE** are given by

$$\text{OPS}_{\text{EMPLOYEE}} = \{\textbf{EMP-ID, EMP-SAL, EMP-RANK, EMP-DEPT}\}.$$

Each own property of a global object is therefore characterized by the set of local properties. Actual derivation of a global property from these local properties is discussed in the Section IV-D.

2) Inherited Properties of Global Object Type: Let $\text{IPS}_{\mathbf{O_L}}$ be the inherited property set of the global object $\mathbf{O_L}$. Consider two objects $\mathbf{O_L}$ and $\mathbf{O_K}$. If $\psi_{\mathbf{L,K}} \cdot \sigma = 1$, then the object $\mathbf{O_L}$ inherits a set of properties from the object $\mathbf{O_K}$. Let $\text{IPS}_{\mathbf{K,L}}$ be the set of properties inherited by the object $\mathbf{O_L}$ from the object $\mathbf{O_K}$.

$$\text{IPS}_{\mathbf{K,L}} = \{\mathbf{P_M} \mid \mathbf{P_M} \in \text{OPS}_{\mathbf{O_K}} \text{ and } \mathbf{P_Y} \in \text{OPS}_{\mathbf{O_L}} \text{ such that not}(\mathbf{P_M} \approx \mathbf{P_Y})\}.$$

Let $\text{OPS}_{\mathbf{FACULTY}}$ be the set {**FAC-ID, FAC-SPECIALIZATION**}. Then, the set $\text{IPS}_{\mathbf{EMPLOYEE,FACULTY}}$ is given by the set {**EMP-SAL, DEPT-NAME, EMP-RANK**}.

An object can have more than one super class. Let $\text{SUP}(\mathbf{O_L})$ be the set of objects having super class relations with $\mathbf{O_L}$. Then the set $\text{SUP}(\mathbf{O_L})$ is given by

$$\text{SUP}(\mathbf{O_L}) = \{\mathbf{O_K} \mid \psi^G_{\mathbf{L,K}\cdot\sigma} = 1\}.$$

If the number of elements in $\text{SUP}(\mathbf{O_L})$ is more than one, then $\mathbf{O_L}$ has more than one super class, and inherits properties from each one of these super classes. In the event of such multiple inheritance, the inherited property set of the object $\mathbf{O_L}$ is derived by computing the union of properties inherited from multiple super classes and then decomposing this union into property equivalence classes. Let $\overline{\text{IPS}_{\mathbf{O_L}}}$ be the set of such property equivalence classes. The set $\overline{\text{IPS}_{\mathbf{O_L}}}$ is given by

$$\overline{\text{IPS}_{\mathbf{O_L}}} = \dot{\bigcup}_{\mathbf{O_K} \in \text{SUP}(\mathbf{O_L})} \text{IPS}_{\mathbf{K,L}}.$$

The set $\text{SUP}(\mathbf{FACULTY})$[2] has **EMPLOYEE** as its element. As such, one can write $\overline{\text{IPS}_{\mathbf{FACULTY}}}$ = {{**EMP-SAL**}, {**DEPT-NAME**}, {**EMP-RANK**}}.

As such, the acquired property set of the object $\mathbf{O_L}$ is given by

$$\text{IPS}_{\mathbf{O_L}} = \{\mathbf{P_X} \mid \mathbf{P_X} \approx [\mathbf{P_N}], [\mathbf{P_N}] \in \overline{\text{IPS}_{\mathbf{O_L}}}\}.$$

Let **FAC-PAY, DEPT-NAME, FAC-RANK** be the set of properties derived from the property equivalence class {**EMP-SAL**}, {**DEPT-NAME**}, and {**EMP-RANK**}. Then, $\text{IPS}_{\mathbf{FACULTY}}$ = {**FAC-PAY, DEPT-NAME, FAC-RANK**}.

Similarly inherited properties of all global objects can be computed. The complete set of properties of the object $\mathbf{O_L}$ denoted by $\text{CPS}_{\mathbf{O_L}}$ is given by

$$\text{CPS}_{\mathbf{O_L}} = \text{OPS}_{\mathbf{O_L}} \bigcup \text{IPS}_{\mathbf{O_L}}.$$

The complete set of properties of object **FACULTY** is given by

$$\text{CPS}_{\mathbf{FACULTY}} = \{\textbf{FAC-ID, FAC-SPECIALIZATION, FAC-PAY, DEPT-NAME, FAC-RANK}\}.$$

This procedure derives global objects and properties of each global object and components of each property of a global object. Unless one fixes the semantic meaning for a global property, it cannot be mapped to its components. A procedure to fix the semantic meaning of a global property is presented in the following subsection.

[2] In some situations this set may have more than one element because of multiple inheritance.

D. Semantic Knowledge of Global Object

Since a global object is characterized by its properties, the meta-properties and meta-values for a global property of a global object are established first.

- **Meta Properties of a Global Property:** Let an integrated property $\mathbf{P_X}$ be derived from an element $[P_x]$ of the set that characterizes $\text{CPS}_{\mathbf{O_L}}$. Note that $\mathbf{P_X}$ must be equivalent to each of its components, and therefore $\mathbf{P_X}$ has the same set of meta-properties as its components. If **DEPT-BUDGET** is the integrated property derived from the property equivalence class {DEPT-BUDGET, DIV-BUDGET}, the components will possess two meta-properties, 'periodicity-of-grant' and 'currency', and **DEPT-BUDGET** will also have the same metaproperties.
- **Meta Values of a Global Property:** The identifier and the meta-values for each metaproperty of the integrated property $\mathbf{P_X}$ are selected from those of any $P_k \in [P_x]$ such that they are acceptable to the majority of global users.

Suppose the meta-values for DEPT-BUDGET and DIV-BUDGET are ('yearly', 'rupees') and ('five years', 'dollars') for the meta-properties ('periodicity-of-budget', 'currency'), respectively. The meta-values 'yearly' and 'dollars' are assigned to the meta-properties 'periodicity-of-budget' and 'currency' respectively for the global property **DEPT-BUDGET**, if and only if they are accepted by the majority of global schema users.

The semantic equation for the global property can be written as follows:

$$SE_{\mathbf{P_K}} = \left\{\{\text{INST}_i(\mathbf{P_K})\}_i, \left\{(M^1_{\mathbf{K}}, V^1_{\mathbf{K}}), (M^2_{\mathbf{K}}, V^2_{\mathbf{K}}), \cdots, \left(M^{|M^{\mathbf{P}}\mathbf{K}|}_{\mathbf{K}}, V^{|M^{\mathbf{P}}\mathbf{K}|}_{\mathbf{K}}\right)\right\}, C_{\mathbf{K}}\right\}.$$

The derivation of instances of global property, INST_i is discussed in Section VI. The derivation of constraints set $C_{\mathbf{K}}$ is covered in [56] and [57].

The semantic knowledge about a global object is given by

$$SE_{\mathbf{O_L}} = \{SE_{\mathbf{P_K}} \mid \mathbf{P_K} \in \text{CPS}_{\mathbf{O_L}}\}.$$

The above equation can be rewritten as

$$SE_{\mathbf{O_L}} = \left\{\{\text{INST}_i(\mathbf{O_L})\}_i, \left\{\left\{(M^1_{\mathbf{K}}, V^1_{\mathbf{K}}), (M^2_{\mathbf{K}}, V^2_{\mathbf{K}}), \cdots, \left(M^{|M^{\mathbf{P}}\mathbf{K}|}_{\mathbf{K}}, V^{|M^{\mathbf{P}}\mathbf{K}|}_{\mathbf{K}}\right)\right\}, C_{\mathbf{K}}\right\}_{\mathbf{K}}\right\}$$

where

$$\text{INST}_i(\mathbf{O_L}) = \{\text{INST}_i(\mathbf{P_K}) \mid \mathbf{P_K} \in \text{CPS}_{\mathbf{O_L}}\}.$$

The first parameter in the equation $SE_{\mathbf{O_L}}$ represents the aggregation of all instances of the object $\mathbf{O_L}$, and the second parameter gives the aggregation of meta-data of all its

properties. Using $SE_{\mathbf{O_L}}$ and the relationship

$$SK_G = \{SE_{\mathbf{O_L}} \mid \mathbf{O_L} \in S_G\}$$

the semantic knowledge of the global schema can be derived.

E. Semantic Mapping of Objects in Global Schema to Componentsin Local Schema

For creating the global schema D_G, so far, we have derived S_G, ψ^G and SK_G. In this section, we derive the mapping knowledge MK_G, which will complete the derivation of the global schema. This mapping knowledge is crucial in the global schema derivation as object integration is achieved by this mapping knowledge.

If two objects belonging to two different local object schemata are declared to be equivalent and are to be integrated, they should be first made compatible. This is achieved by making their equivalent properties semantically compatible, since object integration is manifested through the integration of equivalent properties. As such, it is essential to first identify in how many ways equivalent properties of the entities to be integrated are incompatible. For a group of properties to be integrated, a minimal set of meta-properteis is identified such that the values of these meta-properties are not equal across all properties in the group. Once this set of parameters causing incompatibilities is identified, then these properties can be made compatible with respect to this set of parameters.

In this section, a procedure to map a global property to a property in its equivalence class is presented. This requires the definition of semantic compatibility between two properties.

- **Semantic Compatibility:** For any property $\mathbf{P_K}$, consider the property equivalence class $[P_k]$. Two properties $P_{k'}, P_{k''}$ belonging to $[P_k]$ are said to be compatible with respect to the meta property M^i if and only if $M^i(P_{k'}) = M^i(P_{k''})$, i.e., if and only if $V^i_{k'} = V^i_{k''}$, and the compatibility is denoted by

  $$P_{k'} \overset{M^i}{\sim} P_{k''}$$

  For example, consider two properties, EMP-SAL and FAC-PAY. Assume that EMP-SAL is the monthly salary paid in rupees, and FAC-PAY is the monthly salary paid in dollars. These two properties have meta-value compatibility with respect to PERIODICITY-OF-PAY (say M^1). Therefore, EMP-SAL $\overset{M^1}{\sim}$ FAC-PAY.
- **Transformation Map:** Among the properties of an equivalence class $[P_k]$, if a property $P_{k'}$ is not meta-value compatible with $P_{k''}$ with respect to the meta-property M^j, then it is possible to define a transformation map $t^{M^j}_{P_{k'},P_{k''}}$ which makes $P_{k'}$ meta-value compatible with $P_{k''}$ with respect to the meta-property M^j. Note that $t^{M^j}_{P_{k'},P_{k''}}$ may be a look-up table.

  $$t^{M^j}_{P_{k'},P_{k''}}(P_{k'}) \overset{M^j}{\sim} P_{k''}.$$

  In the above example, FAC-PAY is not compatible with EMP-SAL with respect to the meta-property CURRENCY. The meta-value compatibility can be obtained with the transformation map $t^{\text{CURRENCY}}_{\text{EMP-SAL, FAC-PAY}}$. As such

  $$t^{\text{CURRENCY}}_{\text{EMP-SAL, FAC-PAY}}(\text{EMP-SAL}) \overset{\text{CURRENCY}}{\sim} \text{FAC-PAY}$$

  Here, $t^{\text{CURRENCY}}_{\text{EMP-SAL,FAC-PAY}}(\text{EMP-SAL})$ is $\frac{1}{30}$ times EMP-SAL, assuming \$1 =Rupees 30.
- **Composite Transformation Map:** Two properties $P_{k'}$ and $P_{k''}$ in $[P_k]$ are defined to be semantically compatible with each other if and only if they have meta-value compatibility with respect to all their meta-properties. This is symbolically denoted by $P_{k'} \sim P_{k''}$. Further, if $P_{k'}$ and $P_{k''}$ are not semantically compatible, then a composite transformation map $T_{P_{k'},P_{k''}}$ can potentially be defined which makes $P_{k'}$ sematically compatible with $P_{k''}$.

Suppose $t^1_{P_{k'},P_{k''}}, t^2_{P_{k'},P_{k''}}, \cdots, t^{|M^{P_k}|}_{P_{k'},P_{k''}}$ are the transformation maps which makes $P_{k'}$ meta-value compatible with $P_{k''}$ with respect to the meta properties $M^1, M^2, \cdots, M^{|M^{P_k}|}$ respectively. The composite transformation map can be defined as follows:

$$\begin{aligned} T_{P_{k'},P_{k''}}(P_{k'}) &= \left(t^1_{P_{k'},P_{k''}} \circ t^2_{P_{k'},P_{k''}} \circ \cdots \circ t^{|M^{P_k}|}_{P_{k'},P_{k''}}\right)(P_{k'}) \\ &= t^1_{P_{k'},P_{k''}}\left(t^2_{P_{k'},P_{k''}}\left(\cdots\left(t^{|M^{P_k}|}_{P_{k'},P_{k''}}(P_{k'})\right)\right)\right) \\ &\sim P_{k''}. \end{aligned}$$

Note that if $P_{k'}$ and $P_{k''}$ are already compatible with respect to a particular meta-property, then the corresponding transformation map can be ignored in the construction of the composite transformation map. By using these transformation maps, homogeneity among the component properties can be achieved.

For the two properties **DEPT-BUDGET** and DIV-BUDGET, assume that the meta-properties for these properties are PERIODICITY-OF-BUDGET and CURRENCY. These two properties can be made semantically compatible as shown at the bottom of the page.

We consider another example that involves a lookup table. Consider two properties, DEPT-NAME and DIV-NAME, whose respective sets of distinct values of the instances are {COMPUTER-SCIENCE, MATHEMATICS}

$$\begin{aligned} \textbf{DEPT-BUDGET} &\sim T_{\text{DIV-BUDGET},\textbf{DEPT-BUDGET}}(\text{DIV-BUDGET}) \\ &= t^{\text{PERIODICITY-OF-BUDGET}}_{\text{DIV-BUDGET},\textbf{DEPT-BUDGET}} \circ t^{\text{CURRENCY}}_{\text{DIV-BUDGET},\textbf{DEPT-BUDGET}}(\text{DIV-BUDGET}) \\ &= \left(\frac{1}{5}\right) \cdot (30) \cdot (\text{DIV-BUDGET}). \end{aligned}$$

TABLE I
TRANSFORMATION MAP FOR DIV-NAME

Abbreviated name	Full name
CS	COMPUTER-SCIENCE
MATH	MATHEMATICS

and {CS, MATH}. The corresponding transformation map is shown in Table I.

The semantic compatible class for $\mathbf{P_X}$ denoted by $\mathrm{SCC}(\mathbf{P_X})$ can be defined as follows:

$$\mathrm{SCC}(\mathbf{P_X}) = \{(T_{P_k,\mathbf{P_X}}, P_k) | P_k \in [P_x]\}.$$

The SCC provides the mapping knowledge for a global property. The mapping knowledge for the object $\mathbf{O_L}$ and the entire global schema can be computed using

$$MK_{\mathbf{O_L}} = \{\mathrm{SCC}(\mathbf{P_X}) | \mathbf{P_X} \in PS_{\mathbf{O_L}}\}$$
$$MK_G = \{MK_{\mathbf{O_L}} \mid \mathbf{O_L} \in S_G\}.$$

The procedure described so far applies to objects that fall either under Category I and Category II, which are defined in Section III-B. The integration methodology for other objects is described in the next section.

V. VIEWS ON GLOBAL SCHEMA

Two or more objects in the global schema whose real world states are overlapping but not contained or disjoint can also be integrated through generalization. The generalization process produces a new object called a generalized object. Let $\mathcal{O}_L$ be the generalized object derived from the objects $\mathbf{O_{L_1}}, \mathbf{O_{L_2}}, \cdots, \mathbf{O_{L_n}}$. The real world state of $\mathcal{O}_L$ is the union of the real world states of its components. Since a new object type is added to S_G, the relationship matrix ψ^G needs to be modified accordingly. The properties of $\mathcal{O}_L$ are characterized as follows:

$$\overline{\mathrm{CPS}}_{\mathcal{O}_L} \Leftrightarrow \overset{\bullet}{\bigcap}_{\mathbf{O}_{L_i} \in \mathcal{O}_L} \mathrm{CPS}_{\mathbf{O_L}}$$

where operator $\overset{\bullet}{\bigcap}$ is defined as follows:

$$\mathrm{CPS}_{\mathbf{O_L}} \overset{\bullet}{\bigcap} \mathrm{CPS}_{\mathbf{O_K}} = \Big\{[\mathbf{P_X}] \mid [\mathbf{P_X}] \in \Big\{\mathrm{CPS}_{\mathbf{O_L}} \overset{\bullet}{\bigcup} \mathrm{CPS}_{\mathbf{O_K}}\Big\}, \mathbf{P_M} \in [\mathbf{P_X}], \mathbf{P_N} \in [\mathbf{P_X}], \mathbf{P_M} \in \mathrm{CPS}_{\mathbf{O_L}}, \mathbf{P_N} \in \mathrm{CPS}_{\mathbf{O_K}}\Big\}.$$

The global objects **EMPLOYEE** and **TEMPORARY-STAFF** can be generalized to produce a new object $\mathcal{STAFF}$. Let the set of properties associated with the object **TEMPORARY-STAFF** be **TS-ID**, **WORKING-HOURS**, **WAGES**. Then

$$\overline{\mathrm{CPS}_{\mathcal{STAFF}}} \Leftrightarrow \{\{\textbf{EMP-ID, TS-ID}\}, \{\textbf{EMP-SAL}, \{\textbf{WORKING-HOURS, WAGES}\}\}\}$$
$$\mathrm{CPS}_{\mathcal{O}_L} = \{\mathcal{P}_{\mathcal{X}} \mid \mathcal{P}_{\mathcal{X}} \approx [\mathbf{P_X}], \quad [\mathbf{P_X}] \in \overline{\mathrm{CPS}_{\mathcal{O}_L}}\}.$$

Let $\mathcal{STAFF} - \mathcal{ID}$ and $\mathcal{STAFF} - \mathcal{SAL}$ be the properties of $\mathcal{STAFF}$ derived from the property equivalence classes {**EMP-ID, TS-ID**}, {**EMP-SAL**, {**WORKING-HOURS, WAGES**}} respectively. Then, $\mathrm{CPS}_{\mathcal{STAFF}} = \{\mathcal{STAFF}\text{-}\mathcal{ID}, \mathcal{STAFF}\text{-}\mathcal{SAL}\}$. The semantic knowledge and mapping knowledge of these view objects can be generated as explained in the previous section.

The generation of the global view schema is important because objects in the global schema are generated based on the equivalence classes created by analyzing local object schemata, which does not provide generalization. Such objects may not satisfy the integration needs of global schema users.

For example, the finance department may need to pay both temporary-staff and permanent employees. The view object $\mathcal{STAFF}$ generated by integrating both TEMPORARY-STAFF and EMPLOYEEs is useful to the finance department. Without generalization, the finance department would need to deal separately with **TEMPORARY-STAFF** and **EMPLOYEE**.

The techniques described in Sections IV and V provide a systematic procedure for constructing global schema and for defining views on the global schema. In the following section, the issue of computing instances of objects in the global schema is addressed.

VI. CREATION OF GLOBAL DATABASE

The creation of a global database involves the construction of instances of objects in the global schema.

Let K_l and K_k represent the key properties of two objects O_l and O_k, respectively. Consider jth instance of O_l and ith instance of O_k. Let $K_l(\mathrm{INST}_j(O_l))$ and $K_k(\mathrm{INST}_i(O_k))$ be the values of the key properties $\mathrm{INST}_j(O_l)$ and $\mathrm{INST}_i(O_k)$, respectively. Let T_{K_l,K_k} be the composite transformation map which maps the key values of these instances.

If $T_{K_l,K_k}(K_l(\mathrm{INST}_j(O_l))) = (K_k(\mathrm{INST}_i(O_k)))$, then $\mathrm{INST}_j(O_l)$ is related to $\mathrm{INST}_i(O_k)$, which is symbolically shown as $\mathrm{INST}_j(O_l) \overset{\bullet}{=} \mathrm{INST}_i(O_k)$. For example, if

$$\mathrm{INST}_1(\mathrm{DEPT}) = \{\text{COMPUTER SCIENCE, P. SHAPIRO, 600000}\},$$
$$\mathrm{INST}_2(\mathrm{DEPT}) = \{\text{MATHEMATICS, H. LACEY, 300000}\},$$
$$\mathrm{INST}_1(\mathrm{DIVISION}) = \{\text{MATH, H. LACEY, 50000}\},$$
$$\mathrm{INST}_2(\mathrm{DIVISION}) = \{\text{CS, P. SHAPIRO, 100200}\},$$

then $\mathrm{INST}_1(\mathrm{DEPT}) \overset{\bullet}{=} \mathrm{INST}_2(\mathrm{DIVISION})$. The computation of instances of a global object is based on the concept of object instance equivalence class.

- **Object Instance Equivalence Class**: To compute the instances $\mathbf{O_L}$, consider an object $O_l \overset{*}{\in} \mathbf{O_L}$. Consider the i-th instance of O_l. The object instance equivalence class of an object instance $\text{INST}_i(O_l)$ denoted by $[\text{INST}_i(O_l)]$ is given by

$$[\text{INST}_i(O_l)] = \{\text{INST}_j(O_k) \mid O_k \overset{*}{\in} \mathbf{O_L},\ \text{INST}_i(O_l) \overset{\bullet}{=} \text{INST}_j(O_k)\}.$$

In our example, one gets

$$[\text{INST}_1(\text{DEPT})] = \{\text{INST}_1(\text{DEPT}), \text{INST}_2(\text{DIVISION})\}.$$

Consider an object equivalence class $[\text{INST}_i(O_l)]$. Let the instance of the global object derived from the equivalence class be $\text{INST}_j(\mathbf{O_L})$. This global instance can be alternatively written as

$$\text{INST}_j(\mathbf{O_L}) = \{\text{INST}_j(\mathbf{P_K})\}_{\mathbf{P_K} \in \text{CPS}_{\mathbf{O_L}}}.$$

Suppose the jth instance of the property is to be computed. In the local object schema, $P_x(\text{INST}_j(O_l))$ gives the value of the property P_x of the jth instance of the object O_l. The set of candidate values of the jth instance of the property $\mathbf{P_K}$ is denoted by $[\text{INST}_j(\mathbf{P_K}]$, which represents the property instance equivalence class for the jth instance of $\mathbf{P_K}$:

$$[\text{INST}_j(\mathbf{P_K})] = \{T_{P_x,\mathbf{P_K}}(P_x(\text{INST}_m(O_n))) \mid P_x \in \text{PS}_{O_n},\ P_x \overset{\bullet}{\in} \mathbf{P_K},\ \text{INST}_m(O_n) \overset{\bullet}{=} \text{INST}_i(O_l),\ O_n \overset{\bullet}{\in} \mathbf{O_L}, O_l \overset{\bullet}{\in} \mathbf{O_L}\}.$$

In the example, consider the property **DEPT-BUDGET**.

$$[\text{INST}_1(\textbf{DEPT-BUDGET})] = \{\text{DEPT-BUDGET}(\text{INST}_1(\text{DEPT})), (6)^*\text{DIV-BUDGET}(\text{INST}_2(\text{DIVISION}))\} = \{60000, 601200\}.$$

A selection operator is used to select a correct value for the global property from the above computed candidate values. The selection operator is discussed below.

- **Selection Operator:** In the ideal situation, all the candidate values of a property of a global object should be the same. However, in reality, the candidate values are unequal due to data inconsistencies. As such, each property is assigned a selection operator which selects the appropriate value(s) for the global property instance from the candidate values. A default selection operator can be assigned to each integrated property at the time of design by the global database designer. A user can select a selection operator that suits the application and overwrites the default operator. Let $SO_{\mathbf{P_X}}$ be the selection operator for the property $\mathbf{P_X}$. Then, $\text{INST}_j(\mathbf{P_X}) = SO_{\mathbf{P_X}}([\text{INST}_j(\mathbf{P_X})])$.

Let the selection operator for the property **DEPT-BUDGET** be MAXIMUM. Then, INST_1 (**DEPT-BUDGET**) is given by 601200.

The selection operator resolves data inconsistencies that arise during data integration. One can write the instance of the object $\mathbf{O_L}$ as

$$\text{INST}_i(\mathbf{O_L}) = \{\text{INST}_i(\mathbf{P_X})\}_{\mathbf{P_X} \in \text{CPS}_{\mathbf{O_L}}}.$$

In the example, $\text{INST}_1(\textbf{DEPT}) = \{\text{COMPUTER SCIENCE, P. SHAPIRO, 601200}\}$

This mechanism enables construction of one instance of a global object. Number of instances that can be constructed for $\mathbf{O_L}$ is given by its object instance equivalence closure, which is denoted by $\overline{[\text{INST}(\mathbf{O_L})]}$:

$$\overline{[\text{INST}(\mathbf{O_L})]} = \{[\text{INST}_j(O_k)] \mid O_k \overset{*}{\in} \mathbf{O_L}\}_j.$$

Each instance equivalence class in the above closure corresponds to one instance of the object $\mathbf{O_L}$.

If $\overline{[\text{INST}_1(\textbf{DEPT})]}$ is given by $\{[\text{INST}_1(\text{DEPT})], [\text{INST}_2(\text{DEPT})]\}$, using the two object instance equivalence classes, one can create the instances of **DEPT:**

$$\text{INST}_1(\textbf{DEPT}) = \{\text{COMPUTER SCIENCE, P. SHAPIRO, 601200}\}$$
$$\text{INST}_2(\textbf{DEPT}) = \{\text{MATHEMTICS, H. LACEY, 300000}\}.$$

Using the procedure suggested in this section, instances of any global object can be constructed. Instances of all objects in the global schema collectively form the virtual integrated database. Instances of the view objects can be constructed from this virtual databases.

The methodology presented above provides a step by step procedure for constructing a global schema and for creating a virtual database from existing component database.

VII. Conclusion

The creation of an integrated interface over a given set of existing heterogeneous databases is a challenge faced by many database administrators today. Ideally, the interface should provide a unified view of the data present in the component local databases without requiring the application users to deal with the idiosyncrasies of these local database.

This paper has identified various types of problems that arise during the schema integration process. A methodology has been proposed for the creation of an integrated schema from a given set of local database schema (Sections II–V). This methodology involves acquisition of semantic knowledge pertinent to the objects of a local objects schema. During this knowledge acquisition process, for each property of a local object, parameters that contribute to the semantic meaning of the property are identified (such as meta-properties) and their values (meta-values) are captured. Further, concepts such as object equivalence class and property equivalence class are utilized to facilitate the creation of the integrated schema.

A broad range of semantic conflicts, including scaling conflicts, type conflicts, and level of abstraction conflicts have been addressed in this paper. A number of steps of the schema integration process cannot be automated because of the various semantic incompatibilities present among the component database schemata. The steps in the integration process that can be performed automatically and the steps that require designer's intervention have been distinguished

from each other. A mechanism to check the consistency of the derived global schema has been presented. The proposed methodology derives the global schema and the mapping schema. Using this mapping schema, the global database can be created as described (Section VI) in this paper.

ACKNOWLEDGMENT

The authors would like to thank B. Uma, H. B. Kon, and the anonymous reviewers for their detailed suggestions.

REFERENCES

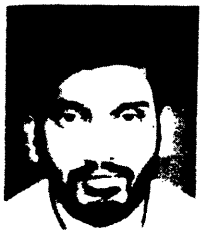
[1] C. Batini and M. Lenzerini, "A methodology for database schema integration in the entity relationship model," *IEEE Trans. Software Eng.* vol. SE-10, no. 6, 1984.
[2] M. Casanova and M. Vidal, "Towards a sound view integration methodology," in *Proc. Second ACM SIGACT/SIGMOD Conf. Principles Database Syst.*, Mar. 1983.
[3] S. B. Yao, "View modelling and integration using the functional datamodel," *IEEE Trans. Software Eng.*, vol. SE-8, no. 6, 1982.
[4] S. B. Navathe and S. G. Gadgil, "A methodology for view integration in logical database design," in *Proc. Eighth Int. Conf. Very Large Data Bases*, 1982.
[5] S. B. Navathe, T. Sashidhar and R. Elmasri, "Relationship merging in schema integration," in *Proc. Tenth Int. Conf. VLDB*, Aug. 1984.
[6] S. B. Navathe, R. Elmasri, and J. Larson, "Integrating user views in database design," *Comput.*, vol. 19, Jan. 1986.
[7] B. Khan, "A structured logical database design methodology," Ph.D. Thesis, Dept. of Comput. Sci., Univ. of Michigan, 1979.
[8] S. Hayne and S. Ram, "Multi-user view integration system (MUVIS): An expert system view integration," in *Proc. Data Eng. Conf.*, 1990.
[9] A. Motro and P. Buneman, "Constructing superviews," in *Proc. Int. Conf. Management Data*, Apr. 1981.
[10] A. Motro, "Superviews: Virtual integration of multiple databases," *IEEE Trans. Software Eng.*, vol. SE-13, no. 7, July 1987.
[11] M. V. Mannino, S. B. Navathe, and W. Effelsberg, "A rule-based approach for merging generalization hierarchies," *Inform. Syst.*, vol. 13, no. 3, 1988.
[12] S. Spaccapietra, "View integration with ERC approach," in *Proc. Workshop Relational Databases Their Extensions*, June 1988.
[13] M. V. Mannino and W. Effelsberg, "A methodology for global schema design," Tech. Rep. TR-84-1, Comput. Inform. Sci. Dept., Univ. of Florida, 1984.
[14] U. Dayal and H. Hwang, "View definition and genralization for database integration in MULTIBASE: A system for heterogeneous distributed databases," *IEEE Trans. Software Eng.*, vol. SE-10, no. 6, 1984.
[15] C. Batini, M. Lenzerini, and S. Navathe, "A comparative analysis of methodologies for database schema integration," *ACM Comput. Surveys*, vol. 18, no. 4, pp. 323–364, 1986.
[16] W. Gotthard, P. C. Lockemann, and A. Neufeld, "System-guided view integration for object-oriented databases," *Knowledge Data Eng.*, vol. 4, no. 1, Feb. 1992.
[17] D. M. Dilts and W. Wu, "Using knowledge-based technology to integrate CIM databases," *IEEE Trans. Knowledge Data Eng.*, vol. 3, no. 2, June 1991.
[18] T. Lander and R. L. Rosenberg, "An overview of MULTIBASE," in *Proc. Second Symp. Distributed Databases*, Sept. 1982.
[19] J. M. Smith *et al.*, "MULTIBASE-Integrating heterogeneous distributed database system," in *Proc. AFIPS*, 1981, vol. 50.
[20] M. Templeton *et al.*, "MERMAID: Experience with network operation," in *Proc. IEEE Int. Conf. Data Eng.*, Feb. 1986.
[21] ——, "MERMAID a front-end to distributed hetrogeneous databases," in *Proc. IEEE*, May 1987.
[22] ——, "Schema integration in MERMAID," in *Position Papers: NSF Workshop Heterogeneous Databases*, Dec. 11–13, 1989.
[23] Y. J. Breitbart and L. R. Tieman, "ADDS-heterogeneous distributed database system," in *Proc. Third Int. Seminar Distributed Database Syst.*, Mar. 1984.
[24] S. M. Deen *et al.*, "The design of a canonical database (PRECI*)," *Comput. J.*, vol. 24, no. 3, 1981.
[25] S. M. Deen, R. R. Amin, and M. C. Taylor, "Query decomposition in PRECI*," in *Proc. Third Int. Seminar Distributed Data Sharing Syst.*, Mar. 1984.
[26] S. M. Deen, R. R. Amin, G. O. Ofori-Dwumfuo and M. C. Taylor, "The architecture of a generalized distributed database system—PRECI*," *Comput. J.*, vol. 28, no. 3, 1985.
[27] S. M. Deen, R. R. Amin, and M. C. Taylor, "Implementation of a prototype for PRECI*," *Comput. J.*, vol. 30, no. 2, 1987.
[28] E. Barkmeyer *et al.*, "An architecture for distributed data management in computer integrated manufacturing," Tech. Rep. NBSIR 86-3312, NBS, Jan. 1986.
[29] V. Krishnamurthy *et al.*, "IMDAS—An integrated manufacturing data administration system," *Data Knowledge Eng.*, vol. 3, 1988.
[30] W. Staniszkis *et al.*, "Architecutre of the network datamanagement systems," in *Proc. 3rd Int. Seminar Distributed Data Sharing Syst.*, Mar. 1984.
[31] W. Litwin and A. Abdellatif, "Multidatabase interoperability," *Comput.*, Dec. 1986.
[32] ——, "An overview of the multidatabase manipulation language MDSL," *Proc. IEEE*, vol. 75, no. 5, pp. 621–631, 1987.
[33] W. Litwin, L. Mark, and N. Roussopoulos, "Interoperability of multiple autonomous databases," *ACM Comput. Surveys*, Sept. 1990.
[34] K. K. Wong and P. Bazex, "MRDSM: A relational multidatabase management system," in *Proc. Third Int. Seminar Distributed Data Sharing Syst.*, Mar. 1984.
[35] "Integrated information support systems report," Tech. Rep. SDS 620140000, ICAM, Materials Lab., Air Force Syst. Command, Wright-Patterson AFB, Feb. 1983.
[36] G. Jakobson, G. Piatetsky-Shapiro, C. Lafond, M. Rajanikanth, and J. Hernandez, "CALIDA: A system for integrated retrieval from multiple heterogeneous databases," Tech. Rep., GTE Lab., 1989.
[37] M. Rajinikanth, G. Jakobson, and G. Piatetsky-Shapiro, "On heterogeneous database integration: One year experience in evaluating CALIDA," in *Proc. Workshop Heterogeneous Databases*, Dec. 1989.
[38] M. Rajinikanth *et al.*, "Multiple database integration in CALIDA: Design and implementation," in *Proc. First Int. Conf. Syst. Integration*, Apr. 1990.
[39] M. P. Reddy, B. E. Prasad, and P. G. Reddy, "Query processing in heterogeneous distributed database management systems," in *Integration of Information Systems: Bridging Heterogeneous Databases*, A. Gupta, Ed. Piscataway, NJ: IEEE, 1989.
[40] Special issue on federated database systems, *Comput.*, vol. 24, no. 12, Dec. 1991.
[41] "Special issue on heterogeneous distributed database management systems," *ACM Comput. Surveys*, vol. 22, no. 3, Sept. 1990.
[42] A. Gupta, Ed., *Integration of Information Systems: Bridging Heterogeneous Database Systems*. Piscataway, NJ: IEEE, 1989.
[43] M. P. Reddy, B. E. Prasad, and P. G. Reddy, "A model for resolving semantic incompatibilities and data inconsistencies in integrating heterogeneous databases," in *Proc. Int. Conf. Management Data*, Dec. 1989.
[44] M. P. Reddy, "Heterogeneous distributed database management systems: Modeling and managing heterogeneous data," Ph.D. Thesis, School of Math. Comput./Inform. Sci., Univ. of Hyderabad, 1990.
[45] R. ElMasri, J. Larson, and S. B. Navathe, "Integration algorithms for federated databases and logical database design," Tech. Rep., Honeywell Corporate Res. Cent., 1987.
[46] R. Y. Wang and S. E. Madnick, "The interdatabase instance identification problem in integrating autonomous systems," in *IEEE Data Eng.*, 1989.
[47] Y. J. Breitbart, P. L. Olson, and G. R. Thompson, "Database integration in a distributed heterogeneous system," in *Proc. IEEE Int. Conf. Data Eng.*, 1986.
[48] N. S. Barghouti and G. E. Kaiser, "Concurrency control in advanced database applications," *ACM Comput. Surveys*, vol. 23, no. 3, Sept. 1991.
[49] C. Date, Ed., *Introduction to Database Systems*. Reading, MA: Addison Wesley, 1990.
[50] E. Bertino *et al.*, "Integration of heterogeneous database applications through an object-oriented interface," *Inform. Syst.*, vol. 14, no. 5, 1989.
[51] S. N. Khoshafian and G. P. Copeland, "Object identity," in , *Readings in Object-Oriented Database Systems* (S. B. Zdonik and D. Maier, Eds.). San Francisco: Morgan Kaufmann, 1990.
[52] J. A. Larson, S. B. Navathe, and R. Elmasri, "A theory of attribute equivalence in databases with application to schema integration," *IEEE Trans. Software Eng.*, vol. 15, no. 4, Apr. 1989.
[53] A. P. Sheth *et al.*, "A tool for integrating conceptual schemas and user views," in *Proc. Fourth Int. Conf. Data Eng.*, Feb. 1980.
[54] J. de Souza, "SIS—A schema integration system," in *Proc. BNCOD5 Conf.*, 1986.
[55] D. A. Simovici and D. C. Stefanescu, "Formal semantics for database schemas," *Inform. Syst.*, vol. 4, no. 11, 1989.

[56] M. P. Reddy, B. E. Prasad, and A. Gupta, "Formulation of global integrity constraints during derivation of global schema," submitted to *IEEE Trans. Knowledge Data Eng.*.

[57] M. P. Reddy, M. Siegel and A. Gupta, "Towards an active schema integration architecture for heterogeneous database systems," in *Proc. Int. Workshop Res. Issues Data Eng.*, 1993.

M. P. Reddy received the Masters degree in physics and the Ph.D. degree in computer science from the University of Hyderabad, Hyderabad, India.

He is currently a Research Associate at the Sloan School of Management, Massachussetts Institute of Technology (MIT). Prior to joining MIT, he worked as an Assistant Professor at the University of Hyderabad. At MIT, he is active in several research projects in the Composite Information Systems Laboratory. He has published several articles in the areas of heterogeneous databases and data quality management. His current research interests include integration of heterogeneous databases, context interchange among heterogeneous information systems, knowledge discovery in databases, and data quality management.



B. E. Prasad received the Ph.D. degree in computer science from the Indian Institute of Technology, Delhi, in 1980 and worked there as a faculty member from 1980 to 1983.

Subsequently, he joined the University of Hyderabad in 1983. From 1985 to 1987, he was a visiting scientist at the Sloan School of Management, Massachusetts Institute of Technology, Cambridge. He is currently Professor and Head of the Department of Computer and Information Sciences, University of Hyderabad, India. He co-edited two books on expert systems for IEEE Press. He has published more than 25 articles and papers on databases and expert systems. His current research interests include object-oriented knowledge base systems, heterogeneous databases, and distributed intelligent systems.

Dr. Prasad received the UNESCO young scientist award in 1987.

P. G. Reddy received the Ph.D. degree from the Indian Institute of Technology, Delhi, India, and subsequently joined the faculty of IIT, Delhi.

In 1983, he joined the University of Hyderabad as Dean for the School of Mathematics and Computer/Information Sciences. During his tenure, he initiated several research projects including a laboratory for aritificial intelligence. He is currently Professor at the Department of Computer and Information Sciences, University of Hyderabad, India. He has published over 100 technical articles in the area of information science, has attended several international and national conferences, and has successfully guided 15 Ph.D. students. His research interests include database systems, expert systems, knowledgebase systems, and computer vision.

Dr. Reddy served as an advisor to the Indian Government to frame a policy for computer education in India. He is a member of board of studies of several Indian universities.

A. Gupta received the Ph.D. degree in computer science from the Indian Institute of Technology, New Delhi, the masters degree in management from the Massachusetts Institute of Technology (MIT), and the Bachelors degree in electrical engineering from the Indian Institute of Technology, Kanpur, India.

He is currently a Co-Director of MIT's Productivity from Information Technology (PROFIT) Initiative and the first and only Senior Research Scientist at the Sloan School of Management of the Massachusetts Institute of Technology. Since joining MIT in 1979, he has been active in the areas of multiprocessor architectures, distributed and heterogeneous database systems, and automated reading of handwritten information. He has published more than 100 technical articles and papers and produced seven books in these areas.

Dr. Gupta serves on the Administrative Committee of the IEEE Industrial Electronics Society and has been assistant chairman for several IECON conferences. He serves as an advisor to a number of leading international organizations.