J. King Saud Univ., Vol. 12, *Comp. & Info. Sci.*, pp. 45-67 (A.H. 1420/2000)

A Semantically Extended Views Integration Method and Its Application to Pavement Views Integration

Rajesh Narang and K. D. Sharma
Department of Computer Science, University of Delhi
Delhi – 110007, India

(Received 20 September 1997; accepted for publication 15 September 1998)

Abstract. Due to increasing interest in distributed databases, the importance of schema integration techniques is significantly increasing. It has been realized database design is such a complex task that it can't be performed in a centralized way, therefore, a more reasonable approach is to first allow the different departments of an organization to build their own schema/view of the database and then integrate them to represent the global schema of the complete knowledge. Keeping this view in mind, an integration strategy based on the concept of structural comparison and semantic comparison of schema is proposed.

The structural comparison finds out similar or near similar types to mainly ascertain subset relationship. It does not analyze the entities very deeply, therefore, it can't detect other relationships hidden in the structural specification of the types. Hence to circumvent these problems, semantic comparison of the types is also considered. The semantic comparison helps to detect other kinds of hidden relationships such as role relationship, identical relationship and compatible relationship between diverse schema components. After this process, all semantic related components are implicitly merged to get a universal view of the knowledge spread in distributed environment. The whole concept is summed up in the form of an integration algorithm.

1. Introduction

Four departments of Central Road Research Institute (CRRI), New Delhi, are conducting research and consultancy work in different structural and functional aspects of pavement design and evaluation. All these departments are independently created and administered. Furthermore different departmental applications also have different view points on the same or similar concepts. Flexible pavement department (FPD) conducts research for new design methods of flexible pavements. Rigid pavement department (RPD) conducts research for new design methods of rigid pavements. Pavement evaluation department (PED) conducts research for methods related to evaluation of existing pavements. Generally, expertise related to pavement evaluation and views corresponding to the expertise are available with PED. Recently scientists working in FPD while working on the design of new flexible methods found more accurate relationship for evaluation of existing flexible pavements and made a view corresponding to it in their department. However, the scientists who are primarily responsible for expertise and views related to pavement evaluation were not aware of it. Therefore, it was felt that only by combining the views of all these applications, will one be able to form an overall picture of complete knowledge distributed in this kind of environment. The present process of integration is manual. In the independent views defined by different departments, it is seen that a reality which is defined as an entity in one view has been represented as an attribute of the entity in another view. In such situations whenever the same reality is represented in different views using different constructs, we say that a structural conflict has occurred. In this paper, we are proposing view integration as the design step aimed at:

- solving problem of structural conflicts in pavement related views defined by pavement engineers.
- producing global view/schema of the database of pavement design method.
- helping the pavement engineers or database administrator (DBA) to capture the whole complexity of data.
- and producing the correct global image of the data structure.

As far as ***related work*** is concerned, various types of approaches in [1; 4] have been proposed for integration of schemas, and they mainly consider structural relationships between entities of different, related and partially overlapping knowledge sources. The conflicts which these approaches broadly consider are of two kinds: naming conflicts and structural conflicts (mainly equivalent, subset, overlapping and disjoint conflicts) on the basis of attribute equivalence. Further, these approaches mainly relate a type (or relationship) of one schema to a type (or relationship) in another schema, and attribute to an attribute to resolve structural conflicts. Our approach influenced by [4] compares these constructs at different levels of perception (type versus relationship, type versus attribute, etc.). The work in [4] does not consider interschema relationships such as "role" relationship and "identical" relationship, and only confines to structural comparison. For example, the million standard axel (**MSA**) load as shown in Fig. 1 type (a) plays the role to find flexible pavement thickness in schema S_1 whereas it is used to find the adjustment in thickness of rigid pavement in RPD. If there is any change in the view corresponding to **MSA** load then this must be updated so that the view remains consistent in both the cases. We have considered such relationships in this paper as a part of semantic comparisons, thus, extending the scope of schema integration. In schema integration process, two schemas are compared at a time to determine the degree of correspondences among inter schema types and conflicts between them. The process of schema integration is divided into two parts, namely, structural comparison and semantic comparison. During structural comparison, the comparison is between (a) a type and a relationship, (b) a relationship and an attribute and (c) a type and an attribute. During semantic comparison relationships are examined deeply to find out whether "identical" relationship or "role" relationship exists between the relationships of two schemas. After this process, all semantic related components are implicitly merged to yield a universal view of the schemas in the different departments. The data model selected is similar to an entity-relationship model.

The schema integration has been designed to achieve the following: (a) automatically integrate elements with or without structural differences, (b) automatically integrate elements with or without semantic differences, and (c) provide an algorithm to perform integration of schemas.

2. Data Model and Concepts of Structural and Semantic Comparisons

The basic construct of the data model are: types, relationships and attributes, and derived constructs are: link and chain. To define each of the construct, first we define an object. Since the process of integration is considered above the level of object , (i.e., at types, relationships, etc.) so we have shown it in italics.

Object

Any real world entity of interest is modeled as an object. An object can represent a pavement, bridge or a vehicle.

Type

An object is modeled as a type if it is perceived as self_existing. In general, let $T_{1j} = \{t_{1ij} \mid I = 1,2,...n\}$ represents the collection of types t_{j1i} belonging to T_{1j} in schema S_j. Here first subscript, 1, in t_{1ij} corresponds to first subscript, 1, of T_{1j} to denote that t_{1ij} belongs to 1st type set T_{1j}. Second subscript, i, in t_{1ij} correspondence to the number of types in T_{1j}. Third subscript, j, in t_{1ij} and second subscript, j, in T_{1j} corresponds to the schema, j, to which t_{1ij} and T_{1j} both belong.

Attribute

The object is considered as an attribute if perceived as a property of some type. Users can define a set of attributes to capture the state of a type. An attribute a_1 is the set of values that a_1 can take. If a_1 is a California bearing ratio (CBR) of a soil then it can take integer values such as 2, 3, 4, etc.,

Link

Suppose t_{111} and t_{121} be the two types in a schema with an attribute a_1 belonging to t_{111} and t_{121}, we say a_1 is a link since it links the two types. It is represented as $t_{111}{*}\, a_1 \leftrightarrow a_1\, {*}t_{121}$. It helps in establishing relationship between t_{111} and t_{121}.

Relationship

An attribute is modeled as a relationship (r_{111}) if perceived as a link between two types. Let $R_{1j} = \{r_{1ij} \mid i = 1,2...n\}$ represent the collection of all relations r_{1ij} in schema S_j. The relationship r_{11i} belonging to R_{11} of schema S_i is a tuple (t_{11i}, t_{12i}) that links type t_{11i} belonging to T_{11} of schema S_i and type t_{12i} belonging to T_{11} of schema S_i. We consider three types of relationships.

(a) Identical relationship

Suppose the relationship, r_{11i} is (t_{11i}, t_{12i}) and relationship r_{12i} is (t_{13i}, t_{14i}). Now if $t_{12i} = t_{14i}$ then interchange t_{12i} and t_{14i} so that r_{11i} becomes (t_{11i}, t_{14i}) and r_{12i} becomes (t_{13i}, t_{12i}). If the two relationships remain valid for all instances and type $t_{11i} \equiv$ type t_{13i} we say that the two relationships r_{11i} and r_{12i} are identical.

(b) Role relationship

Suppose the relationship, r_{11i} is (t_{11i}, t_{12i}) and relationship r_{12i} is (t_{13i}, t_{14i}). Now if t_{12i} is not $\equiv t_{14i}$ then interchange t_{12i} and t_{i14} so that r_{11i} becomes (t_{11i}, t_{14i}) and r_{12i} becomes (t_{13i}, t_{12i}). If the two relationships remain valid for all instances and type $t_{11i} \equiv$ type t_{13i} we say that the role relationship exists between two types t_{11i} and t_{13i} and two relationships r_{11i} and r_{12i} are compatible.

(c) Incompatible relationship

Suppose the relationship, r_{11i} is (t_{11i}, t_{12i}) and relationship r_{12i} is (t_{13i}, t_{14i}). Now if t_{12i} is not $\equiv t_{14i}$ and and type t_{11i} is not $\equiv$ type t_{13i} we say that the two relationships r_{11i} and r_{12i} are incompatible.

Chain

Suppose $t_{11j}, t_{12j}, ... t_{1nj}$ is an orderly sequence of types in a given schema j such that for all i in (1, 2, 3...n), t_{1ij} is linked to t_{1i+1j} by an attribute then $t_{11j} \rightarrow t_{12j} ... \rightarrow t_{1ij} \rightarrow t_{1i+1j}$ and $t_{1ij} \rightarrow t_{1i+1j} \rightarrow ... t_{1nj}$ are the two chains. The types or attributes or relationships can be mixed in one chain.

We define *cardinality* as a count of number of types (or relationships or attributes) in a chain. Let $t_{11j} \rightarrow t_{12j}. \rightarrow t_{1ij} \rightarrow t_{1i+1j}$ and $t_{1ij} \rightarrow t_{1i+1j} \rightarrow .t_{1nj}$ be the two chains. we say the chain $t_{11j} \rightarrow t_{12j}. \rightarrow t_{1ij} \rightarrow t_{1i}+1_j$ is a longer chain. If cardinality ($t_{11j} \rightarrow t_{12j} ... \rightarrow t_{1ij} \rightarrow t_{1i+1j}$) > cardinality ($t_{1ij} \rightarrow t_{1i+1j} \rightarrow t_{1nj}$).

It is assumed that in the distributed environment, knowledge of each department is expressed in the form of schema as follows:

$S_j = (T_{ij}, R_{ij})$

Let $D_{1j} = (d_{1ij})$ and $D'_{1j+1} = (d'_{1ij+1})$ where d_{1ij} denotes the destination types in S_j that have direct links with types of T_{1j}

d'_{1ij+1} denotes the set of types in S_{j+1} which have structural relationship with types in set D_{1j} of S_j

Example 1: Flexible pavement (**FP**) is an example of **type** with attributes soil_subgrade, sub_base_course, base_course and surface_course.

Example 2: Suppose t_{111} belonging to T_{11} is a type which contains all the necessary inputs required to compute **MSA** load (refer type (a) Fig. 1a). Let t_{121} belonging to T_{11} be another type with attributes: CBR value (cbr_val), msa and tot(al)_thickness (refer type (b) Fig. (1a). Here types t_{111} and t_{121} can be linked together through the attribute msa to get view of total_thickness for various values of CBR. Hence the attribute msa is a **link** for the two types. It is represented as t_{111} * msa ↔ msa * t_{121}.

Type traffic_volume		**Type fp_thickness**	
(com_veh:	float;		
Design_life:	float;	(cbr_val:	float;
r_val:	float;	msa:	float;
g_rate:	float;	tot_thickness:	float;
t_elapsed:	float;		
l_type:	float;		
r_type:	float;		
Cbr_val:	float;		
Msa:	float);		

Type (a): t_{111} Type(b): t_{121}

r_{111} : compute_fp_thickness(traffic_volume, fp_thickness)

Fig. 1a. Partial overview of S_1.

2.1 Knowledge-form consulted for establishing interschema relationship

Knowledge required for integrating two views at a time is expressed in the form of declarative statements known as correspondence assertion (CA). It is provided by the user or DBA. The CA conveys that data structure in one schema is somehow related to the semantics of some piece of data structure in another view of the schema. Firstly, structural comparison is performed using CAs, if no correspondence can be established between two schemas the elements of two schemas are not forwarded to the semantic comparators for deep analysis. Examples 3 and 4 demonstrate how the knowledge is expressed in the form of correspondence assertion and is used to integrate two schemas/views.

2.2. Structural comparator

Structural comparison singles out the set of components in the different knowledge sources that present some degree of structural connectivity. It determines common and uncommon structural characteristics and correspondences between individual types in different sources. The result of this comparison reveals several sorts of inter type structural correspondences.

2.2.1 Element correspondence assertions

Let x_{11i}, x_{11j} be two elements (i.e., types, relationships or attributes). Let x_{11i} belonging to S_1 and x_{11j} belonging to S_j. The following assertions will be used to establish correspondence between x_{11i} and x_{11j} **denoted by x_{11i} (cor) x_{11j}**

(a) Element equivalence assertion

Two elements (types or attributes) x_{11i} and x_{11j} in different schemas S_i and S_j are called equivalent if the same modeling constructs and perceptions are applied, and no incoherence enters into the specifications even if their respective names are different. In general, the element assertion that x_{11i} and x_{11j} are equivalent is defined as follows: $x_{11i} \equiv x_{11j}$ states that x_{11i} and x_{11j} have same attributes.

(b) Element disjoint assertion

The assertion that x_{11i} and x_{11j} are disjoint is expressed as follows x_{11i} is not $\equiv x_{11j}$ means x_{11i} and x_{11j} have no attribute in common. In other words x_{11i} n $x_{11j}=\phi$. To consider structural comparison we discuss the following:

Step 1: Equivalence of two elements

(a) Equivalence of two types

Let t_{111} be a type in S_1 and t_{112} be a type in S_2 with the following CA:
$a_{11} = a_{21}$ and $a_{12} = a_{22},\ldots, a_{1n} = a_{2n}$

Then it $\Rightarrow$ $t_{111} \equiv t_{112}$. The integrated type t_{11I} in integrated schema (IS) corresponding to t_{111} and t_{112} will have an attribute ai for each attribute correspondence $a_{1i} = a_{2i}$ $\forall$ i. The CA in such cases will be represented as: $t_{111} \equiv t_{112}$ with corresponding attributes: $a_{11} = a_{21}$, $a_{12} = a_{22},\ldots, a_{1n} = a_{2n}$ Let t_{111} be a type in S_1 and t_{112} be another type in S_2 with CA: $a_{11} = a_{21}$, $a_{12} = a_{22},\ldots, a_{1n-1} = a_{2n-1}$. The attributes of t_{111} are $a_{11}, a_{12}, \ldots, a_{1n}$ and attributes of t_{112} are $a_{21}, a_{22}, \ldots a_{2n-1}$.

The integrated type t_{11I} in IS corresponding to t_{111} and t_{112} will have: (i) an attribute ai corresponding to each attribute correspondence $a_{1i} = a_{2i}$ $\forall$ i in (i = 1, 2,..., n-1) and (ii) an attribute a_j (in the present case it is a_n) for each attribute of t_{111} that has no corresponding attribute in t_{112}.

(b) Equivalence of two relationships

Let r_{111} be a relationship in S_1, r_{112} be a relationship in S_2 with corresponding attributes: $a_{11} = a_{21}$, $a_{12} = a_{22}, \ldots, a_{1n} = a_{2n}$. Then the integrated relationship r_{11I} in IS corresponding to r_{111} and r_{112} will have an attribute corresponding to each $a_{1i} = a_{2i}$ for all i in (i = 1,2, ...n). The correspondence relationship for two equivalent relationships is represented as $r_{111} = r_{112}$ with corresponding attributes $a_{11} = a_{21}, a_{12} = a_{22}, \ldots, a_{1n} = a_{2n}$. Refer Fig. 1b.

(c) Equivalence of a relationship and a type

Let t_{111} be a type in S_1 and r_{111} be a relationship in S_2 with corresponding attributes $a_{11} = a_{21}$, $a_{12} = a_{22},\ldots a_{1n}= a_{2n}$. Then the integration of t_{111} and r_{111} generates a type t_{11I} with an attribute ai for each attribute corresponding to $a_{1i} = a_{2i}$. The CA for an equivalent type t_{111} and a relationship r_{111} will be represented as $r_{111} \equiv t_{111}$ with attributes $a_{11} = a_{21}, a_{12} = a_{22}, \ldots a_{1n} = a_{2n}$. Refer Fig. 1c.

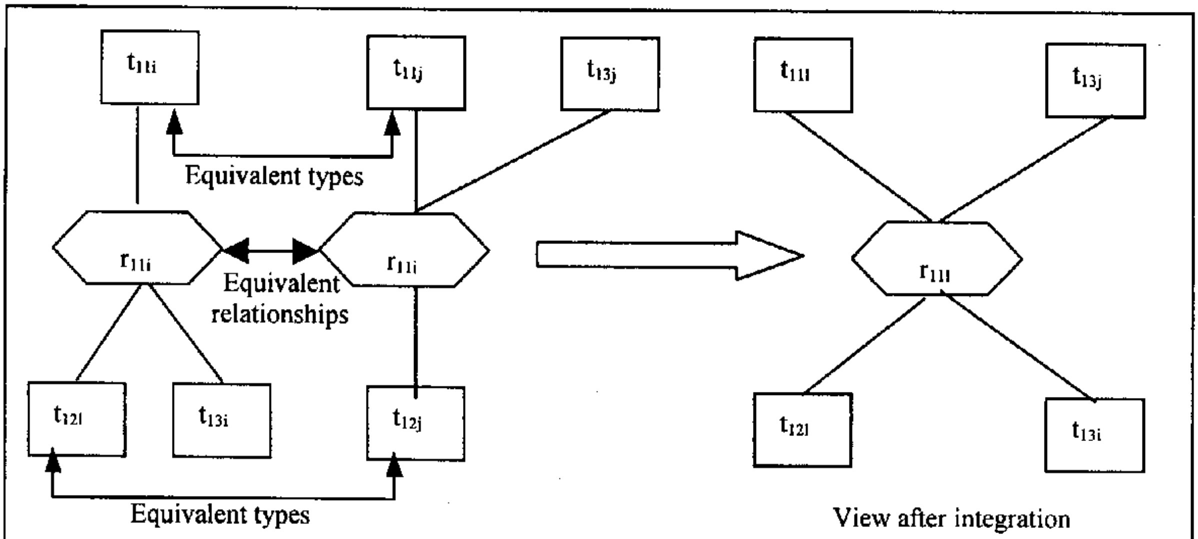

Fig. 1b. Integration of two equivalent relationships r11j and r11j.

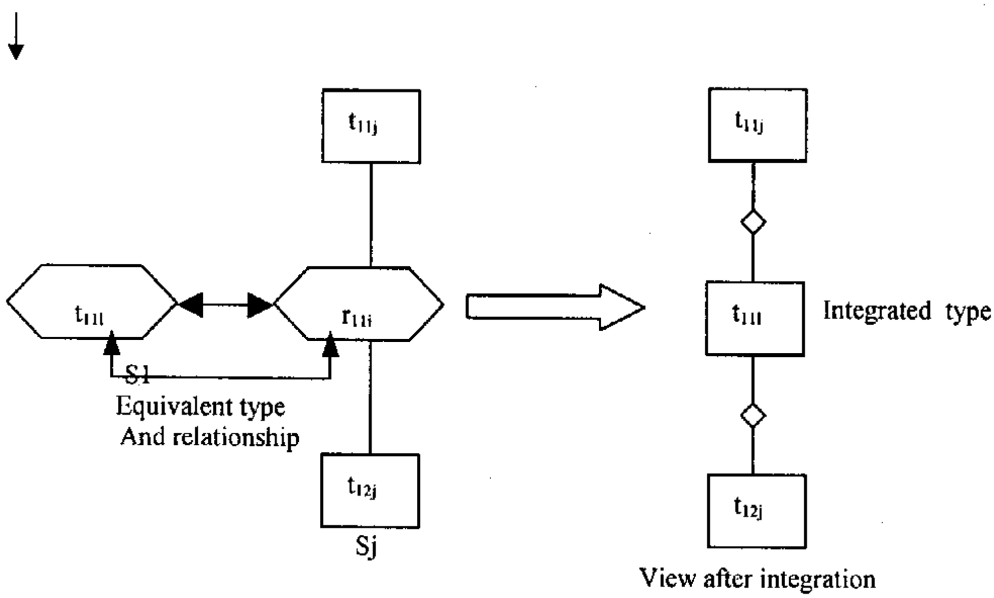

Fig. 1c. Integration of a type and a relationship.

Step 2: Addition of noncorresponding elements

Any element (type or relationship) that exists in one schema and has no corresponding element in any other schema is added to the integrated schema with all its attributes without modification.

Step 3: Integration of two chains terminated with equivalent relationship
Suppose r_{111} in S_1 and r_{112} in S_2 are two equivalent relationships such that r_{111} links x_{111}, x_{121}, ..., x_{1n1}, r_{112} links y_{112}, y_{122}, y_{1n2} and $x_{111} = y_{112}$, $x_{121} = y_{122}$, In such cases, we add the assertion $x_{111} \rightarrow r_{111} = y_{112} \rightarrow r_{112}$, $x_{121} \rightarrow r_{111} = y_{122} \rightarrow r_{112}$,... to the set of CA.

Step 4: Chains integration
We categorize a chain either a short chain or a long chain. If there are only two elements in a chain we call it a short chain. If there are more than two elements in a chain we call it a long chain. Here we discuss three cases.

(a) Integration of two short chains
Let x_{111} and y_{121} be two elements forming a chain in s_1, x_{112} and Y_{122} be two elements forming a chain in S_2. The following CAs are given:

$x_{111} = x_{112}$, $y_{121} = y_{122}$ ... (1)

$x_{111} \rightarrow y_{121} = x_{112} \rightarrow y_{122}$ ... (2)

CA (1) $\Rightarrow$ x_{111} and x_{112}, and y_{121} and y_{122} can be integrated.
Let x_{111} be the integrated element in IS corresponding to x_{111} and x_{112}. Let y_{111} be the integrated element in IS corresponding to y_{121} and y_{122}. Integration of chains $x_{111} \rightarrow y_{121}$ and $x_{112} \rightarrow y_{122}$ will give us a chain $x_{111} \rightarrow y_{11i}$ in IS . It will be an attribute chain if x_{111} is a type and y_{111} is an attribute.
We examine the above case later in more details during semantic comparison. The cases considered are:

(i) x_{111} not $\equiv$ x_{112} but $y_{121} = y_{122}$

(ii) $x_{111} = x_{112}$ but y_{121} not $\equiv$ y_{122}

(b) Integration of a short chain and a long chain
Suppose the short chain adopts the direct way to go from some specific source information to a target information in one step in S_1. On the other hand, let long chain adopts the indirect way to go from same source information to the same destination information in S_2. That is, we have $x_{111} \rightarrow x_{1n1} = y_{112} \rightarrow y_{122} \rightarrow .y_{1p2}$ ... (3) and

$x_{111} = y_{112}$ and $x_{1n1} = y_{1p2}$ ...(4)

In this case, the long chain will be selected for integration into the IS. The integration proceeds as follows:

CA (4) $\Rightarrow$ x_{111} and y_{112}, and x_{1n1} and y_{1p2} can be integrated.
Let z_{111} be the integrated element in IS corresponding to x_{111} and y_{112}. Let z_{1n1} be the integrated element in IS corresponding to x_{1n1} and y_{1p2}. In CA (3), a short chain and a long chain are involved. As indicated above, we select the long chain, namely, $z_{111} \rightarrow y'_{122} \rightarrow y'_{132} \rightarrow .z_{1nI}$ where z_{111} and z_{1nI} are the integrated elements obtained from CA (4). y'_{122}, y'_{132}, ..., $y'_{1p-1,2}$ are elements corresponding to y_{122}, y_{132}, ..., $y_{1p-1,2}$ respectively.

(c) Integration of two long chains
Let the corresponding assertion between two chains be

$x_{111} \rightarrow x_{121} \rightarrow ... \rightarrow x_{1n1} = y_{112} \rightarrow y_{122} \rightarrow .y_{1p2}$ ...(5)

with $x_{111} = y_{112}$ ...(6) and
$x_{1n1} = y_{1p2}$ ...(7).

CA (6) $\Rightarrow$ x_{111} and y_{112} can be integrated.

CA (7) $\Rightarrow$ x_{1n1} and y_{1p2} can be integrated.

Let z_{11I} be the integrated element corresponding to integration of x_{111} and y_{112}. Let z_{1pI} be the integrated element corresponding to integration of x_{1n1} and y_{1p2} There will be two long chains in the IS for CA (5). The chains will be:

$z_{11I} \rightarrow x'_{121}, ..., x'_{1n-11} \rightarrow z_{1pI}$

$z_{11I} \rightarrow y'_{112}, ..., y'_{1p-12} \rightarrow z_{1pI}$

z_{11I} and z_{1pI} are the integrated elements obtained from CA (6) and (7) respectively in both the chains. $x'_{121}, ..., x'_{1n-11}$ are elements of IS corresponding to $x_{121} \rightarrow ... \rightarrow x_{1n-11}$ and $y'_{112}, ..., y'_{1p-12}$ are elements of IS corresponding to $y_{112}, ..., y_{1p-12}$

Suppose neither chain is short chain and both the long chains record alternative ways to go from the same (or corresponding) source information to the same (or corresponding) destination information. If two long chains do not have common elements then both chains have to be integrated into the IS.

Step 5: Integration of chains terminating with attributes

Let $x_{111}, x_{121},...,x_{1n1}$ be the elements and 'a' be an attribute in S_1. Let $y_{112}, y_{1n2}..., y_{1n2}$ be the elements and 'b' be an attribute in S_2 with following CA : $x_{111} = y_{112}$ ... (8) and $a = b$ ... (9)

Let z_{11I} be the integrated element in IS corresponding to x_{111} and y_{112}. Let there be another corresponding assertion,

$x_{111} * a = y_{112} \rightarrow y_{122}, ..., y_{1n2} * b$ ... (10)

(* indicates that element after it is an attribute in the chain) (8), (9) and (10) together generate in IS an attribute b' which is an attribute of y'_{1n2} where y'_{1n2} is an element corresponding to y_{1n2}. If the correspondence assertion is of the form:

$x_{111} \rightarrow x_{121}, ..., x_{1n1} * a = y_{112} \rightarrow y_{122}, ..., y_{1p2} * b$ ... (11)

(8), (9) and (11) together generate in IS two attributes: a' and b'. a' is an attribute of x'_{1n1} where x'_{1n1} is an element corresponding to x_{1n1} in IS. b' is an attribute of y'_{1p2} where y'_{1p2} is an element corresponding to y_{1p2} in IS.

2.3 Semantic comparator

It finds out additional semantics between types which are hidden in structured specifications of different sources and can't be ascertained by the structure based comparison. For identification of such associations types of one schema which associates type in different knowledge sources are checked.

We detect following three types of relationships during semantic comparison.

- **Identical** means two relationships have exactly the same semantics in the real world although they may have different names.

- **Role** relationship means same type plays two different roles in two different schemas.
- **Incompatible** means two relationships have no common properties.

2.3.1 Semantic comparison

We perform semantic comparisons in three steps. The diagrammatic view of semantic comparison is shown in Fig. 2.

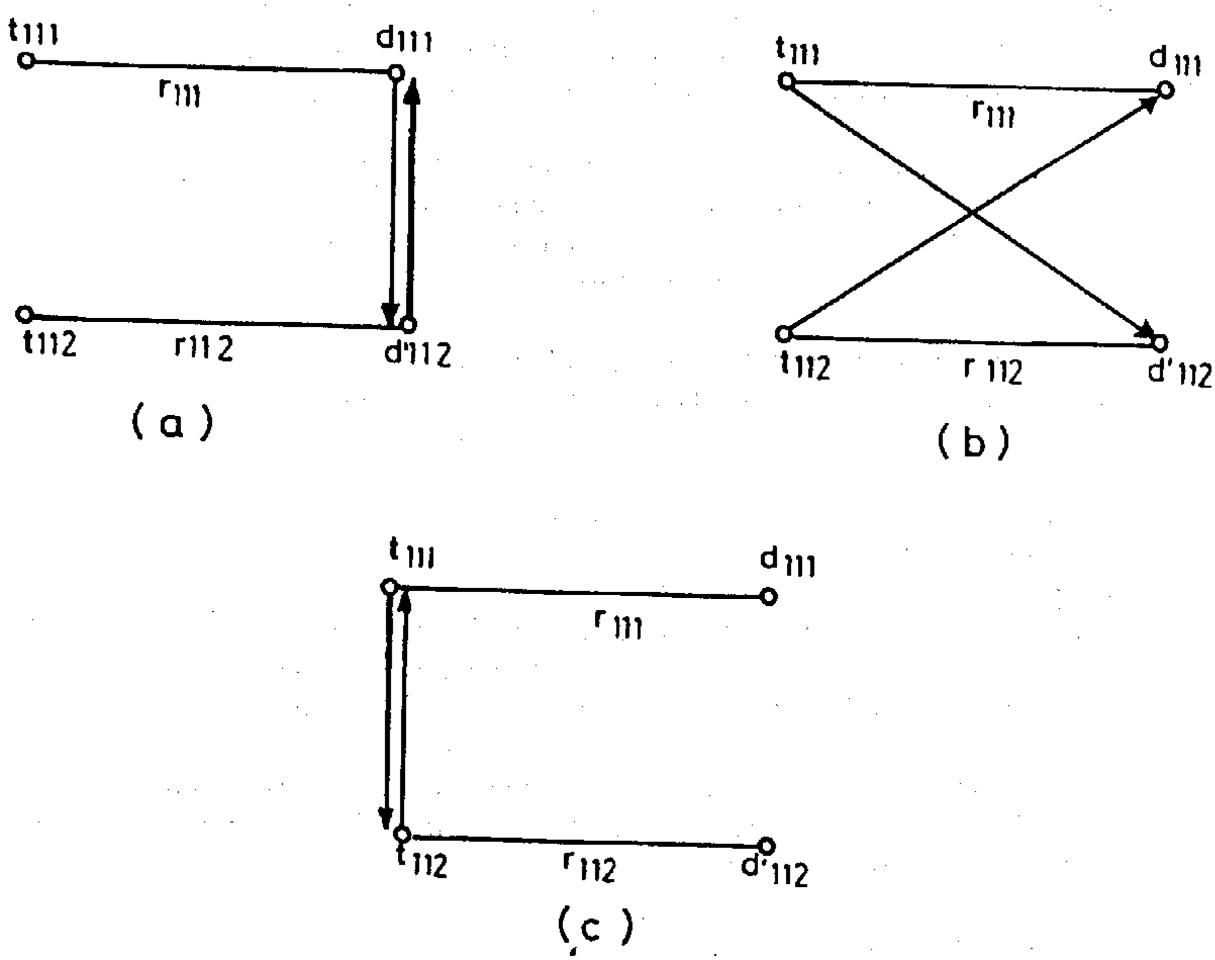

Fig. 2. Semantic comparison.

Step 1:

This step is divided into following sub steps.

(a) For all types t_{111} in T_{11} in S_1, find set of destination types d_{1i1} which have direct associations with this particular type t_{111}. Put all types d_{1i1} in set D_{11} so that $D_{11} = \{d_{1i1} \mid i=1, 2, ..., n\}$. Pick up one of the types from D_{11} say d_{111} which has a relation r_{111} with type t_{111} of T_{11} in schema S_1. Thus r_{111} is r_{111} (t_{111}, d_{111}).

(b) Next, in S_2 from set T_{k2}, find types t_{ki2} (i = 1,2, ..., n) which have structural relationship with types of D_{11} and rename these types as d'_{1i2}. Let $D'_{12} = \{d'_{1i2}\}$ for i = 1, 2, ..., n. From D'_{12}, pick up one type, say, d'_{112} which has a specific structural relationship with d_{111}. Finally in S_2 from say, T_{12}, pick up a type, say, t_{112} which has a relationship r_{112} with d'_{112} so that r_{112} is r_{112} (t_{112}, d'_{112}).

(c) To establish semantic similarities, compare all four types t_{111}, d_{111}, t_{112}, d'_{112} and the two relationships r_{111}, r_{112} in two steps.

(d) Replace d_{111} with d'_{112} in r_{111} (t_{111}, d_{111}) so that it becomes r_{111} (t_{111}, d'_{112}).

(e) Replace d'_{112} with d_{111} in r_{112} (t_{111}, d'_{112}) so that it becomes r_{112} (t_{111}, d_{111}).

Step 2:

```
If (d111 ≡ d'112) (refer Fig. 2a) then
if ( r111 (t111, d'112 ) ∧ r112 ( t112, d111) remain valid for all instances ) (refer Fig. 2b) then
        if ( t111 ≡ t112 ) (refer Fig. 2c) then
                • relationships r111 and r112 are identical
                endif
                if ( t111 not ≡ t112 ) then
                there exists a role relationship between d111 and d'112.
                • relationship r111 and r112 are compatible.
                endif
        endif
endif
```

Step 3:

```
If (d111 not ≡ d'112) (refer Fig. 2a) then
if ( r111 (t111, d'112) ∧ r112 (t112, d111) remain valid for all instances) (refer Fig. 2b) then
                if (t111 ≡ t112) (refer Fig. 2c) then
                • there exists a role relationship between t111 and t112.
                • relationship r111 and r112 are compatible.
            endif
                if ( t111 not ≡ t112 ) then
                • relationship r111 and r112 are incompatible,
                 no assumption can be made about any type
                endif
        endif
endif
```

endfor

2.4 Integration algorithm

Input : Two schemas S_1 and S_2 and the CA in between S_1 and S_2.

Output: An IS.

Steps: Five steps have been defined for integrating S_1 and S_2.

Step 1:

For all elements correspondence assertion between x_{111} and x_{112}

if ($x_{111} \wedge x_{112}$ both are types) then

- execute Step (1a) of section 2.2.1

endif

if ($x_{111} \wedge x_{112}$ both are relationships) then

- execute Step (1b) of section 2.2.1

endif

- Mark x_{111} in S_1 and x_{112} in S_2 as processed

endfor

Step 2:

For all elements (in S_1 and S_2) that has not been processed at step (1) in S_1 and S_2

- execute Step (2) of section 2.2.1 . That is, step for adding non corresponding elements.
- Mark the added element and its attributes as processed.

endfor

Step 3:

For all pairs of relationship r_{111} in S_1 and r_{112} in S_2

if ((r_{111} links $x_{111}, x_{121}, \ldots, x_{1n1}$) $\wedge$

(r_{112} links $y_{112}, y_{122}, \ldots, y_{1n2}$) $\wedge$

($r_{112} = r_{112}, x_{111} = y_{112}, x_{121} = y_{122}, \ldots, x_{1n1} = y_{1n2}$)) then

- Add to the set of chain CAs the assertion:

$x_{111} \rightarrow r_{111} = \quad y_{112} \rightarrow r_{112}, x_{121} \rightarrow r_{111} = y_{122} \rightarrow r_{112}, \ldots$

(i.e., Step (3) of section 2.2.1)

endif

endfor

Step 4:

For all chain CA $x_{111} \rightarrow x_{112} \rightarrow x_{1n1} = y_{112} \rightarrow y_{122} \rightarrow y_{1n2}$

if (two short chains are involved) then

- execute Step (4) of section 2.2.1

endif

if (a long chain and a short chains are involved) then

- execute Step (4) of section 2.2.1

endif

if (two long chains are involved) then

- execute Step (4) of section 2.2.1

endif

- Mark the corresponding chains as processed in S_1 and S_2.

endfor
Step 5:
For all attribute CA: $a_1 = a_2$
if (a chain CA is terminated with attributes a_1 and a_2) then
- execute Step (5) of section 2.2.1

else
- add both attributes a_1 and a_2 in IS

endif
endfor
Step 6:
Refinement of IS
Input: An IS
Output: An equivalent IS'
Steps: **For** all (type t_{111} linked to destination type d_{111} and type t_{112} linked to destination type d'_{112})
- perform semantic comparison as given in section 2.3.1

endfor

3. The Application of Structural Comparison

The examples that have been chosen to show the feasibility of aforesaid algorithm are from the pavement design and evaluation environment.
An overview of design, evaluation and overlay thickness estimation of pavements is presented in appendix-1.
Example 3: Let us consider the case of evaluating the structural adequacy of a rigid pavement. Two independent views for it have been created in schema S_1 and S_2 (refer Fig. 3a). The task here is to combine the two views to form an integrated view.
The set of CA between S_1 and S_2, given as an input, consists of following attributes:
Rigid_pavement = Rigid_pavement with corresponding attributes:
section_name = section_name and
existing_slab_thickness = existing_slab_thickness ... (12)
Traffic_effect = Evaluation with corresponding attributes
structural_adequacy = structural_adequacy ...(13)
Overlay_design = overlay with corresponding attributes
Pavement_condition# = Pavement_condition# ... (14)
Traffic_effect*Adjusted_slab_thickness = This_traffic_volume * Adjusted_thickness ...(15)
Traffic_effect*Adjusted_slab_thickness = Evaluation → This_traffic_volume * Adjusted_thickness ...(16)
Rigid_pavement → Traffic_effect = Rigid_pavement → Is_subject_to → This_traffic_volume→ Evaluation ... (17)

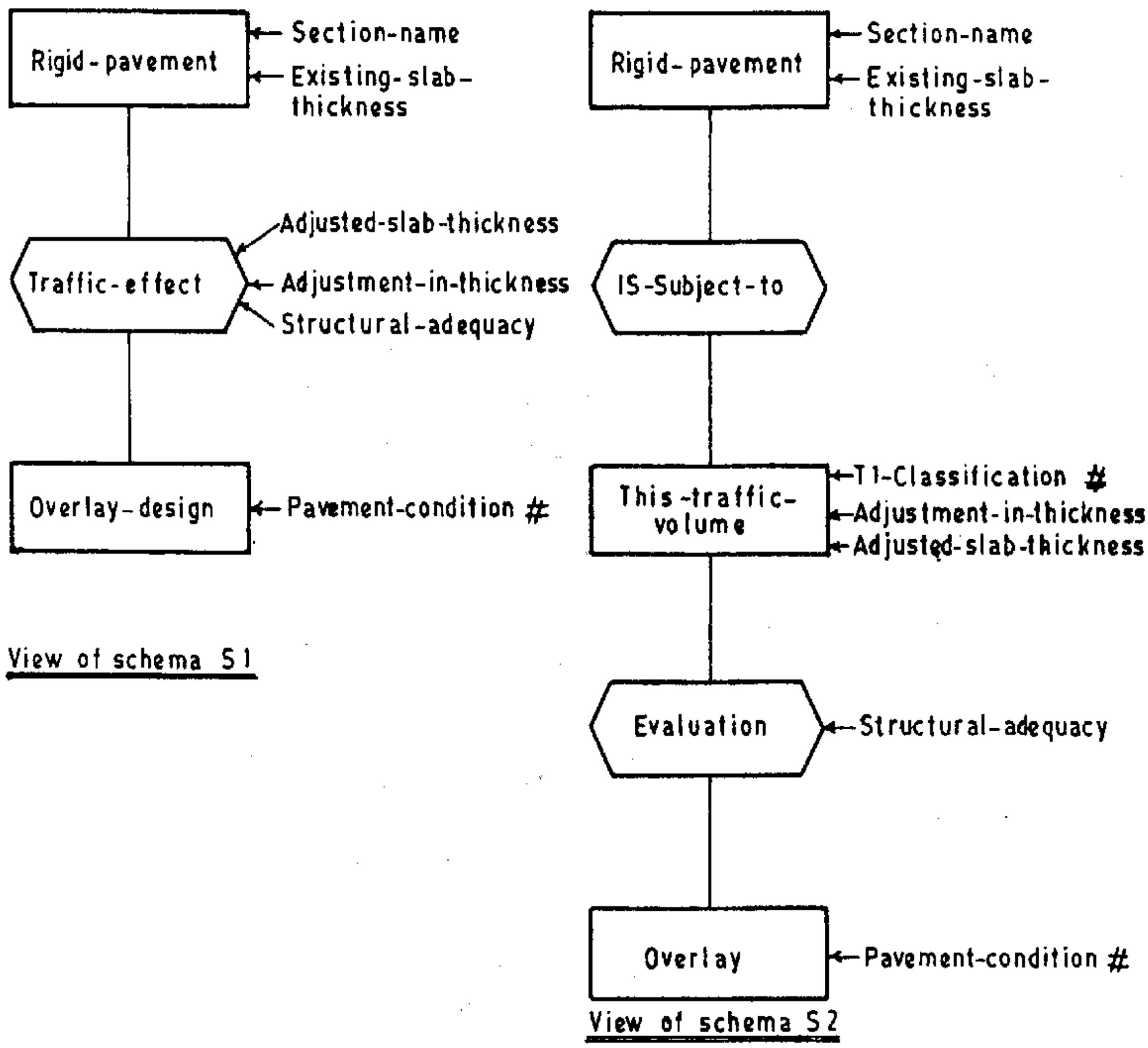

Fig. 3a. View to compute overlay design of rigid pavement in schema S1 and S2.

The CA (15) and (16) will be taken up at the end since attributes are involved in them. Based on CA (11) to (17), we perform structural comparison on the basis of first five steps outlined in the Integration Algorithm.

Step 1:

(a) As type Rigid_pavement of S_1 and type Rigid_pavement of S_2 are equivalent (CA 12), so as per step (1a), we insert a type Rigid_pavement in IS with attributes: section_name and existing_slab_thickness. Mark correspondence assertion (12), type Rigid_pavement with attributes, section_name and existing_slab_thic-kness in S_1 and S_2 as marked.

(b) As relationship Traffic_effect of S_1 = relationship, Evaluation of S_2 (CA 13) with attribute, structural_adequacy so as per step (1b), we add a relationship, say, Evaluation with attribute structural_adequacy in IS. Mark CA (13), relationship, Traffic_effect in S_1 and relationship Evaluation in S_2 as processed.

(c) The type, Overlay_design = type, Overlay, with attribute, Pavement_condition# (CA 14) so we add a type, say, Overlay_design, with attribute, Pavement_condition# in IS. Mark correspondence assertion (15), type, Overlay_design in S_1 and type, Overlay in S_2 as processed.

Step 2:

We select elements that exist in S_1 and have no corresponding element in other schema S_2. We find none. Next, We select elements that exist in S_1 and have no corresponding element in other schema S_2. We find there exist, a relationship, Is_subject_to, in S_2 without any attribute. We add it in IS. Further, we find a type, This_traffic_volume, with attributes, TI_Classification# and Adjusted_thickness. The attribute Adjusted_thickness is under consideration in correspondence assertion (15) and (16), so, we add the type, This_traffic_volume, with only one attribute, TI_classification# in IS.

Step 3:

In S_1, Overlay_design is linked to Traffic_effect, i.e., we have Overlay_design → Traffic_effect. In S_2, Overlay is linked to Evaluation, i.e., we have Overlay→ Evaluation (refer Fig. 3a). The CA

(13) ⇒ relationships, Traffic_effect and Evaluation are equivalent. The CA (14) ⇒ types, Overlay_design and Overlay are equivalent. Hence as per step 3, we get the following new CA.

Overlay_design → Traffic_effect = Overlay → Evaluation ... (18)

As CAs (15) and (16) involve attributes, therefore, they are put aside and will be taken up at the end. The CA (17) has two chains:

(i) Rigid_pavement → Traffic_effect and

(ii) Rigid_pavement →Is_subject_to→ This_traffic_volume → Evaluation.

As latter chain is a longer chain, therefore, we select it for integration into IS. The CA (18) has two short chains. We can select either. Suppose we decide to select chain, Overlay_design → Traffic_effect. The type, Overlay_design is all right as it correspondence to the type that has already been selected in the IS. Out of relationships, Traffic_ effect and Evaluation, we select the relationship, Evaluation, as Evaluation is already selected for integration in IS.

Step 4:

We take up the CAs (15) and (16) which were put aside. The CA (15) is Traffic_effect* Adjusted_slab_thickness = This_traffic_volume*Adjusted_thickness

The CA (16) is

Traffic_effect * Adjusted_slab_thickness = Evaluation →This_traffic_volume * Adjusted_thickness

From CA (13), we have Traffic_effect = Evaluation. The CA (13) and (16) and step (5) with CA (14) ⇒ an attribute, adjusted_thickness, be generated in IS. This attribute is the attribute of type, This_traffic_volume. The integrated view is shown in Fig. 3b.

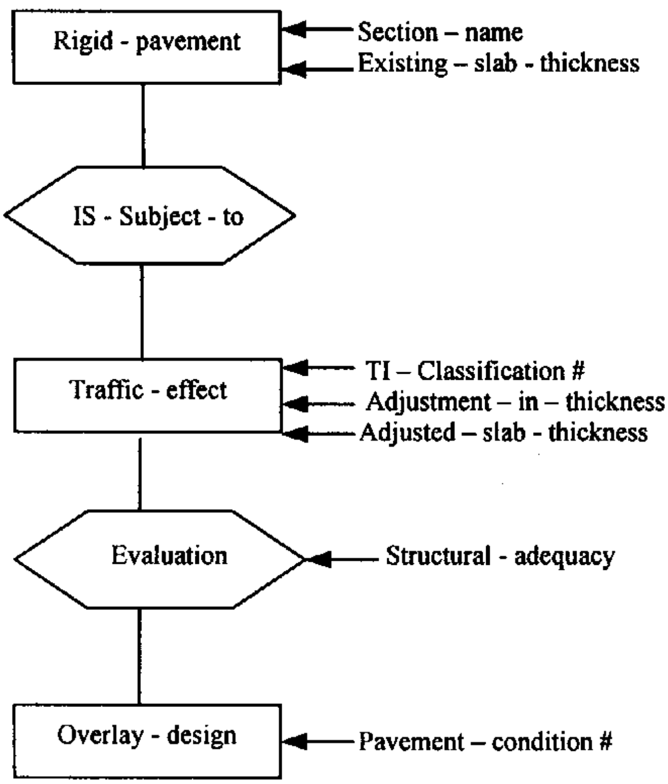

Fig. 3b. View to compute overlay design after integrating view of S_1 and S_2.

Output

The integrated view shown in Fig. 3b is a complete view for all the departments of CRRI. If they use it further updation and changes will give them a consistent view on *overlay design method of rigid pavement.*

4. Application of Semantic Comparison

Indian Road Congress (IRC) has published the guidelines for the design of FP based on the concept of cumulative MSA Loads known as CBR method. The types for FP design and overlay thickness estimation are shown in Fig. 4 as schema S_2. In this section application of semantic comparison is shown, to find out different relationships between the types of schema S_1 and S_2. We first consider structure based comparisons and then semantic based comparison.

Type traffic_intensity		**Type deflection**	
(com_veh:	float;	(no._of_def_val	float;
Design_life:	float;	def_val():	float;
g_rate:	float;	avg_def:	float;
t_elapsed:	float;	std_dev:	float;)
l_type:	float;		
r_type:	float;		
msa:	float;		
cbr_val:	float;)		
Type (a): t_{112}		Type (b): t_{122}	
Type ch_deflection		Type corrections	
(avg_def:	float;	(temp:	float;
std_def:	float;	moist_cont:	float;
temp_corr:	float;	rainfall:	float;
moist_corr:	float;)	plascity:	float;
		temp_corr:	float;
		moist_corr:	float);
Type(c): t_{132}		Type (d): t_{142}	
Type allowable_deflection			
(msa:	float;		
Allowed_def:	float;)		
Type (e) : t_{15}			

Fig. 4. Partial overview of S_2.

Example 4: In Fig. 4, we represent the relationship, compute_allowable_deflection (traffic_intensity, allowable_deflection) as $r_{112}(t_{112}, t_{152})$.
CA between S_1 (refer Fig. 1a) and S_2 (refer Fig. 4) consists of traffic_intensity * msa $\leftrightarrow$ msa * allowable_deflection ...(19).
In Fig. 1a, $T_{11} = \{t_{111}, t_{121}\}$. In Fig. 4, $T_{12} = \{t_{112}, t_{122}, t_{132}, t_{142}, t_{152}\}$.
The semantic comparison as per step 6 proceeds in three steps.

Step 1:
This step consists of following sub steps:
(a) For all types t_{111} in T_{11}, we find all the types which have direct relations with t_{111}. As there is only one type t_{121} in T_{11}, the attribute of t_{111} and t_{121} are compared. It is found 'msa' is a common attribute between t_{111} and t_{121}. The types t_{111} and t_{121} can be linked

together through attribute 'msa' (as given in CA (19)). Hence t_{121} is the type which has a direct link with t_{111}. Hence $D_{11} = \{d_{111} = t_{121}\}$.
In S_1, we pick up a type that has a relation with d_{111}. The type is t_{111} since t_{111} (= traffic_volume) and d_{111} (= fp_thickness) have a common attribute 'msa'. As per CA (19), the two types can be linked together to form a relation r_{111} (= compute_fp_thickness). Hence the relation got is compute_fp_thickness (traffic_volume, fp_thickness). It is represented as $\boldsymbol{r_{111}\ (t_{111}, d_{111})}$.
(b) In S_2, $T_{12} = \{t_{112}, t_{122}, t_{132}, t_{142}, t_{152}\}$. We try to find all types, t_{1j2}, j = 1, 2, 3, 4, 5 which have relationship with types of D_{11}, i.e., with d_{111}.

- d_{111} and t_{112} have two attributes: 'cbr_val' and 'msa' in common.
- d_{111} and t_{122} are disjoint types
- d_{111} and t_{132} are disjoint types
- d_{111} and t_{142} are disjoint types

d_{111} and t_{152} have a common attribute msa. So both can be linked together (CA (1)) .Thus $D'_{12} = \{t_{112}, t_{152}\}$. If we select t_{112}, it is compared with type t_{111} which we have associated with d_{111} in r_{111}. We find t_{111} and t_{112} are equivalent, hence there is no point in associating it again with d_{111}. Therefore, we select the other type t_{152} and retain it in D'_{12}. So $D'_{12} = \{t_{152} = d'_{112}\}$

Finally we select a type t_{1j2}, j = 1, 2, 3, 4, 5 from T_{12} in S_2 which has a structural relationship with d'_{112}. As before, d'_{112} (= t_{152}) is compared with all type of T_{12} on the basis of attribute comparisons.

It is established d'_{112} (= t_{152}) is only related with type t_{112}. The relationship between t_{112} and d'_{112} is named as compute_allowable_deflection. The relationship is compute_allowable_deflection (traffic_intensity, allowable_deflection). It is represented as $\boldsymbol{r_{112}\ (t_{112}, d'_{112})}$

(c) We have found four types : t_{111}, d_{111}, t_{112}, d'_{112} and two relationships : r_{111} (t_{111}, d_{111}) and r_{112} (t_{112}, d'_{112}).
(d) Replace d_{111} in r_{111} with d'_{112} so r_{111} (t_{111}, d_{111}) becomes r_{111} (t_{111}, d'_{112}). The relationship is compute_fp_thickness (traffic_volume, allowable_deflection).
(e) Similarly replace d'_{112} in r_{112} with d_{111} so that r_{112} (t_{112}, d'_{112}) becomes r_{112} (t_{112}, d_{111}).The relationship is compute_allowable_deflection (traffic_intensity, fp_thickness).

Step 2:
We compare types d_{111} and d'_{112} and find d_{111} is not $\equiv d'_{112}$. The comparison is done on the basis of attribute equivalence. Thus this step is skipped.

Step 3:
As d_{111} is not $\equiv d'_{112}$:
r_{111} (t_{111}, d'_{112}) and r_{112} (t_{112}, d_{111}) remain valid for all the instances.

As $t_{111} \equiv t_{112}$,

- there exists a role relationship between types t_{111} and t_{112}.
- type t_{111} (= traffic_volume) plays the role of finding d_{111} (= fp_thickness) in S_1 type t_{112} (=
- traffic_intensity) plays the role of finding d'_{112} (= allowable deflection) in S_2. Although types t_{111} and t_{112} are equivalent but different emphasis is placed on its usage in S_1 and S_2.
- r_{111} and r_{112} are compatible relations.

As ($t_{111} \equiv t_{112}$) so this condition is evaluated as false so we skip this step. With this, the process of semantic analysis comes to an end as there is no more type in T_{11}.

Output

- Semantic comparison has shown that there is a role relationship between type t_{111} (traffic_volume in Fig,. 1a) and type t_{112} (traffic_intensity in Fig. 4). However, they play different roles in two schema S_1 and S_2
- Two relationships r_{111} (compute_fp_thickness) and r_{112} (compute_allowable_deflection) are compatible since one of the two types used in each of the two relations is common.

5. Summary

The strength of the proposed algorithm is its integration of both structural and semantic comparison into a single schema integration model. It can be used by the pavement engineers or DBA during view integration phase. Its usage has helped in accumulating knowledge and describing integrated views about different aspects of pavement design, evaluation and overlay. This wealth of pavement knowledge is being utilized by the different divisions of CRRI in design and code of softwares which they are developing using different methods of pavement design and evaluation on whenever required basis.

The schema integration process proposed here is summed up in the form of an algorithm. It semiautomates the process of generating a database schema. Though it is not fully automatic yet it can act as a helpful assistant to a DBA. There is a reason to suppose that the fully automated algorithm is not possible. The algorithm can be applied to database design in two phases. The initial phase involves constructing a new database manually. As the database increases in complexity, the algorithm begins to assist the DBA by identifying relationships between new data elements and existing ones.

References

[1] Howard W. Beck, Tarek Anwar and Shamkant B. Navathe. " A Conceptual Clustering Algorithm for Database Schema Design." *IEEE Transaction on Knowledge and Data Engineering,* 6, No. 3 (1994).

[2] Martin Odell. *Object Oriented Analysis and Design.* Prentice Hall, 1991.

[3] Qing Li and Mcleod D. "Conceptual Database Evolution Through Learning in Object Databases. " *IEEE Transaction on Knowledge and Data Engineering,* 6, No. 2 (1994), 205-222.

[4] Stefano Spaccapietra and Christine Parent. " View integration: A Step Forward in Solving Structural Conflicts. " *IEEE Transaction on Knowledge and Data Engineering,* 6, No. 2 (1994), 258-273.

[5] IRC:58 - Guidelines. *Design of Rigid Pavement for Highways.* Indian Road Congress, 1998.

[6] IRC:37 - Guidelines. *Guidelines for the Design of Flexible Pavement.* Indian Road Congress, 1984.

[7] SP:17 - Guidelines. *Recommendations About Overlays on Cement Concrete Pavements.* Indian Road Congress, 1977.

[8] IRC:76 - *Guidelines. Tentative Guidelines for Structural Strength Evaluation of Rigid Airfield Pavements.* Indian Road Congress, 1979.

APPENDIX -1

List of notations and terms used

(a) Notations used

1.	Schema j	S_j
2.	T_{1j}	1st set of types in schema j
3.	t_{1ij}	ith element (i.e., type) $\varepsilon\ T_{1j}$
4.	R_{1j}	1st set of relations in schema j
5.	r_{1ij}	ith element (i.e., relation) $\varepsilon\ R_{1j}$
6.	D_{1j}	1st set of destination types in Sj that have direct links with types of set T_{1j}.
7	$D'_{1j}+1$	1st set of types in S_{j+1} that have structural relationship with types of set D_{1j} of S_j
8.	D_a	Allowable Deflection
9.	D_c	Characteristic Deflection
10.	h_o	Overlay thickness
11.	Chain between two elements	$x_{111} \rightarrow x_{112}$, where x_{111} and x_{112} are elements (i.e., type or relationship)
12.	Chain between an element and an attribute	x_{111} * a where x_{111} is an element and a is an attribute.
13.	$\equiv$	Equivalent to
14.	$\Rightarrow$	Implies
15.	$\leftrightarrow$	Link between two types

(b) Terms used

There are two types of pavements : (i) rigid pavements (RPs) and (ii) flexible pavements (FPs).

1) Rigid pavements : The Rigid pavements are those which possess note worthy flexural strength and one of its design methods known as Design Wheel Method is based on (i) foundation strength, (ii) flexural strength, (iii) design wheel load, (iv) traffic intensity (TI) and (v) critical stress condition.

Traffic intensity of a pavement for 20 years is predicted as: $T = p(1+r)^{n+20}$ ....(1e)

Where T = design traffic intensity in terms of commercial vehicles/day,

p = traffic intensity at last count,

r = annual rate of increase of traffic intensity,

n = number of years since last traffic count and commissioning the new pavement.

Depending on the value of T a pavement is classified as A, B, C, D, E, F, G type of pavement, and based on it a due adjustment in thickness of pavement is made as (refer Table 1).

Table 1. Adjustment in pavement thickness for given values of T

Pavement Classification (TI-Classification#)	T	Adjustment in thickness of Pavement (in cm) (h_t)
A	0 to 15	-5
B	15 to 45	-5
...	...	...
G	4500	2

Source: Ref. [8]

Table 2 . Criteria to categorize pavement condition

Pavement Condition#	Length of crack in meter/10 sq.meter	Category
1	0 to 1.0	sound
2	Exceeding 1.0 upto 2.5	slightly cracked
3	Exceeding 2.5 upto 5.5	fairly cracked
4	Exceeding 5.5 upto 8.5	moderately cracked
5	Exceeding 8.5 upto 12.0	badly cracked
6	Exceeding 12.0	very badly cracked

Source: Ref. [10]

2)Flexible pavement: The pavements which possess neglible flexural strength are called flexible pavements.

3) Overlay design of flexible pavement: The evaluation of existing pavement can be done using a method known as Benkleman Beam Deflection (BBD) method. BBD values are generally adjusted to account for the effects of pavement temperature, rainfall, and soil moisture condition on the magnitude of the characteristic deflection (D_c). The overlay thickness (h_o) required is determined after finding the allowable deflection (D_a) in the pavement under the design load. The D_a is found on the basis of MSA load.

4) California bearing ratio: It is used to measure the strength of soil.

طريقة موسعة مبنية على مقارنة المعاني لتكمل واجهات قواعد البيانات الموزعة وتطبيقاتها

راجيش نازنج وك. دي. شرما
قسم علوم الحاسب، جامعة دهلي
دلهي - ١١٠٠٠٧ الهند

(قدّم للنشر في ٢٠/٩/١٩٩٧م ؛ وقبل للنشر في ١٥/٩/١٩٩٨م)

ملخص البحث. نظراً للاهتمام المتزايد بقواعد البيانات الموزعة، ازداد اهتمام الباحثين بأساليب تكامل وربط تشكيلات قواعد البيانات. ولعله من الواضح الآن أن تصميم قواعد البيانات الموزعة أمرٌ بالغ التعقيد ومن الصعب التحكم في عملية التصميم بصورة مركزية. ومن ثم يكون من الأفضل أن يسمح للأقسام المختلفة في المؤسسات والهيئات بتصميم تشكيلات وواجهات قواعد البيانات الخاصة بها، ثم تجميع وربط هذه التشكيلات/ الواجهات الفرعية لبناء التشكيل الكلي والواجهة الكلية لقاعدة البيانات. هذه الورقة تقترح استراتيجية تكاملية لقواعد البيانات الموزعة مبنية على فكرة المقارنة التركيبية ومقارنة المعاني.

المقارنة التركيبية تكتشف الأنواع المتماثلة وشبه المتماثلة إلا زنها لا تجري تحليلاً عميقاً للمكونات، ومن ثم يصعب عليها اكتشاف العلاقات الأخرى المبنية على التوصيف الداخلي للأنواع. للتغلب على هذه المشكلة نأخذ في الاعتبار عملية مقارنة المعاني. تساعد مقارنة المعاني على اكتشاف العلاقات الخفية مثل: علاقات الأدوار، علاقات التطابق وعلاقات التوافق بين وحدات التشكيلات المتنوعة. بعد عملية مقارنة المعاني يتم تجميع وربط كل وحدات التشكيلات التي توجد بينها روابط معنوية للحصول على صورة متكاملة للمعرفة المخزنة في قاعدة البيانات الموزعة. الفكرة بكاملها تلخص في النهاية في شكل خوارزمية تكامل.