DETECTING HETEROGENEITY IN A MULTIDATABASE ENVIRONMENT THROUGH AN O-O MODEL

Danielle Boulanger, Joël Colloc

DETECTING HETEROGENEITY IN A MULTIDATABASE ENVIRONMENT THROUGH AN O-O MODEL

D. BOULANGER, J. COLLOC

Centre de Recherche I.A.E. - URA CNRS 1257
Université Jean-Moulin - LYON 3
15, Quai Claude Bernard
69007 LYON - France

INTRODUCTION

It becomes imperative to provide new functionalities to support the interoperability of autonomous database systems without requiring their total global integration.
Several approaches to allow cooperation between automated information systems have been proposed aiming at different levels of integration (DAYA 84) (MOTR 87) :
. "multidatabase" approach (LITW 90) in which users query different bases with a single order but they must specify the location of data ;
. "federated" approach which supports location transparency (SHET 90). When databases have to interoperate, a data model must be chosen as Canonical Data Model (CDM) for the federation.
System consisting of multiple databases may be characterized along three dimensional spaces defined by distribution, heterogeneity and autonomy (SHET 90). We only focus on heterogeneity.
Our presentation is organized as follows. Section 1 presents an heterogeneity typology. Section 2 discusses some features of a data model that make it suitable as a CDM in a federation. Section 3 describes our model, develops the concept of application object type and presents algorithms which detect structural incompatibility and semantic incompatibility.

1. HETEROGENEITY

We check many types of heterogeneities : those due to differences in DBMSs and those due to the differences in the semantics of data. In the heterogeneous environment, we have to solve data inconsistencies that exist in different data sources (BREIT 90). We can also distinguish between two types of incompatibility : structural and semantic.

2. REPRESENTATION SUITABILITY OF A CANONICAL DATA MODEL

2.1. Criteria

The representation suitability of a database is involved in its data model (structures, operations and integrity constraints) and can be determined by two criteria :

. **semantic capability** is the degree to which the model can represent a conceptualization (BROD 84) (SALT 91) ;

. **semantic relativism** translates the multiplicity of possible representations of a given real world. Supporting semantic relativism is one solution to cope with conflicting representations which may be detected among various pre-existing schemas, if we will not modify them (to respect the autonomy of native systems).

2.2. Impact of cooperation environment

The use of a common CDM solves the problem of syntactic heterogeneities provided by the existence of different local data models.

IFIP DS-5 Semantics of Interoperable Database Systems, Lorne, Victoria, Australia 16-20 November 1992 Preprint Version of "Resolving Heterogeneity in a Multidatabase Environment through an O-O Model"

2.2.1. Need of knowledge acquisition process
This process yields explicit all the extracted semantics in a completed representation of each database. The rich semantic capability of the CDM allows more semantic information in the component schemas than in the relating local schemas (we use the reference architecture described by (BOUL 90) (SHET 90)).

2.2.2. Semantic capability
A CDM should be closed with respect to : classification/instantiation, generalization/specialization, aggregation/decomposition, definition of encapsulated new operations and integrity constraints for extending the behavioural capability of the model.

3. AN OBJECT ORIENTED MODEL

In our model (COLL 89) (COLL90), the objects are nested in each other. According to the applications, a unit corresponding the reference object type (or level) has to be defined : each instance of this type is called **unique object.**
This notion defines a boundary between the **internal** and **external** level : all the component objects belong to the internal environment, inside the "unique object", while the others reside in the external environment. Objects that contribute to build the unique object are called sub-objects.
The internal structure represents the content of the object and its composing sub-objects which determine a partition of set of properties.
The external environment expresses the relationships of an object with the others.
Each sub-object can be seen, at any time, as a unique object. Its proper sub-objects build up its internal structure. We call **zoom effect** this adapted perception of objects. The model provides a twin conceptual level.

3.1. The internal level or "inner object"

It includes :
- aggregation relationships of sub-objects (IS-PART-OF)
- relationships grouping all the attributes of an object (HAS-A).

Aggregation relationships establish a multiple upward inheritance which passes composing sub-objects properties on to "compound object".
The internal level encapsulates attributes and functions which represent the static characteristics and the behaviour of the objects.
The constraints on the objects are either static (constraints on the structures, the attributes, the cardinalities) or dynamic ; in this case, they are functions of time.
The spontaneous evolution of the object state is translated by dynamical functions of the inner object. These functions automatically modify the composition links and the reference values of the attributes, at any time.
Evaluation functions of the "inner object" compare the structure of an object (on its composing sub-objects), the value of particular attributes to these of the other objects.

3.2. The external level or outer-object

It includes the generalization/specialization relationships (IS-A) between object types and object sub-types. These relationships establish a type hierarchy and simple descending inheritance which passes object type properties on to its sub-types.

The evaluation functions belonging to instances of the application object type, assess all the object type instances involved in an application.

The model makes a distinction between type concept (object structure and properties) and class concept (a set of instances).

3.3. Links between external level and internal level

3.3.1. Instantiation
The internal and external levels are linked by the **instantiation mechanism** which determines new objects in accordance with each type.
The **qualifier attributes** are particular attributes (listed at the external level as characteristics of the object type) whose instantiation is mandatory at the internal level. All attributes of an object are not "qualifier attributes". In this case, the value attribution is optional.
Communication between objects is performed by **messages** and **interface functions.** Messages carry information from an emitter object to one or several receiver objects.
Message may play the role of negotiation protocols (JOHA 89).
The validation of the access conditions to receiver objects is performed by the **inner** interface functions.

Note the methods encapsulated in the objects are translated by functions, in our model.

3.3.2. Application object type
The application object type is a predefined system type, whose aim is to describe applications which suit with several user needs.
Objects of application type encapsulate :
. data - metaattributes are used to represent intermediate states and results of a treatment ;
. metafunctions are designed to define existing relationships, else than composition, which exists between domain objects of some types, involved in an application. They belong to the internal environment of application objects.
Metafunctions send messages to obtain the internal state of these objects and store the received result in metaattributes.

Advantages of object of application type :
First, application objects have the same nature as domain objects. However their structures are driven by a preestablished system type and metafunctions are written using a metalanguage.
Lemma - application objects have same properties as other objects - they use encapsulations with modularity features. Application objects can be involved as sub-objects in a composition hierarchy to build a new application objects. So, metaattributes and metafunctions of application sub-objects are inherited by the built in application object using multiple upward inner inheritance.

Evaluation functions make use of partial similitude algorithms.

This O-O Model offers semantic capability, the application object type concept allows to implement similarity algorithms (section 4) which provide semantic relativism.

3.4. Using application object to treat interoperable query in an heterogeneous environment

The application object type provides the structure of application instances and the needed system functions to allow the database administrator to create, delete and update them.
Object application metafunctions provide a view according to some user's needs. This concept allows to represent external, federated and export schemas.
We use the concept of object application type to treat interoperable query in an heterogeneous environment. In (BOUL 92), we present a cooperation between two preexisting relational databases homogeneous by the model, heterogeneous by the DBMSs.

4. OBJECT SIMILARITY ALGORITHMS

The model allows to find common points between objects, first by comparing their types and second by using evaluation functions which return values representing the similarity degree of their contents.
Evaluation functions compare the structure of an object with the structure of others. Evaluation functions provide a solution for uncertainty threshold by object classification and distance computing (FU 83) (COLL 87) (MANA 88).
The whole process involves in turn generalization/specialization object composition and object instantiation.

We distinguish two kinds of similarity providing several levels of successive refinements.
In a first step, the comparison process tries to make out global or partial similarity between objects. If successful, the second step searches for a value resemblance between "qualifier attributes" of similar objects.
We focus on the first step.

Structural similarity has to establish that two objects posses similar sub-objects and so on recursively towards the leaves of the composition hierarchy of the two objects. Global or partial structural similarity between objects takes place by comparing the lists of their sub-object types.
Sometimes, only a partial structure similarity involving several sub-objects can be found. As some parts of objects are more important than others, the partial similarity can be very valuable (for example, the lock is the major sub-object door).
If we compare OBJ1 and OBJ2 structures whose respective types are TYPE1 and TYPE2, several cases can occur :

Case 1 : OBJ1 and OBJ2 belong to the same type.
OBJ1 and OBJ2 have, by definition, a strong degree of structural relationship. But we have to consider that the structural relationship depends on the specialization level brought by the two object types.

Case 2 : OBJ1 and OBJ2 belong to the same type but have heterogeneous structures
The sub-objects that compose the object can be selected among several listed distinct object types giving heterogeneous structures.
To compare the heterogeneous structures of OBJ1 and OBJ2, we have to build an evaluation function which will be included in an application object.
The evaluation function gets two lists of sub-object types. By comparing the two lists, the evaluation function computes a structural similarity score taking into account the relative importance of sub-objects.

Case 3 : OBJ1 and OBJ2 belong two different types TYPE1 and TYPE2.
First we must check if one of the two types is a super-type of the other one.

Case 3 : *first variant* - TYPE1 is a TYPE2 super-type or conversely
Type1 and TYPE2 are connected by an inclusion dependence and OBJ1 and OBJ2 are both TYPE1 instances (the most general type). Thus, if an evaluation function exists for TYPE1, we can use it to compare OBJ1 and OBJ2.

Case 3 : *second variant* : TYPE1 and TYPE2 are independent
We can assume that OBJ1 and OBJ2 show a poor similarity degree. However, we can imagine that these objects have been built independently at different times, by several users. So, we have to verify if they have a partial similitude structure. We use an evaluation function to examine sub-object types involved in OBJ1 and OBJ2 respective structures.

This evaluation function belongs to every object types and is context free. The function implements a recursive comparison algorithm.

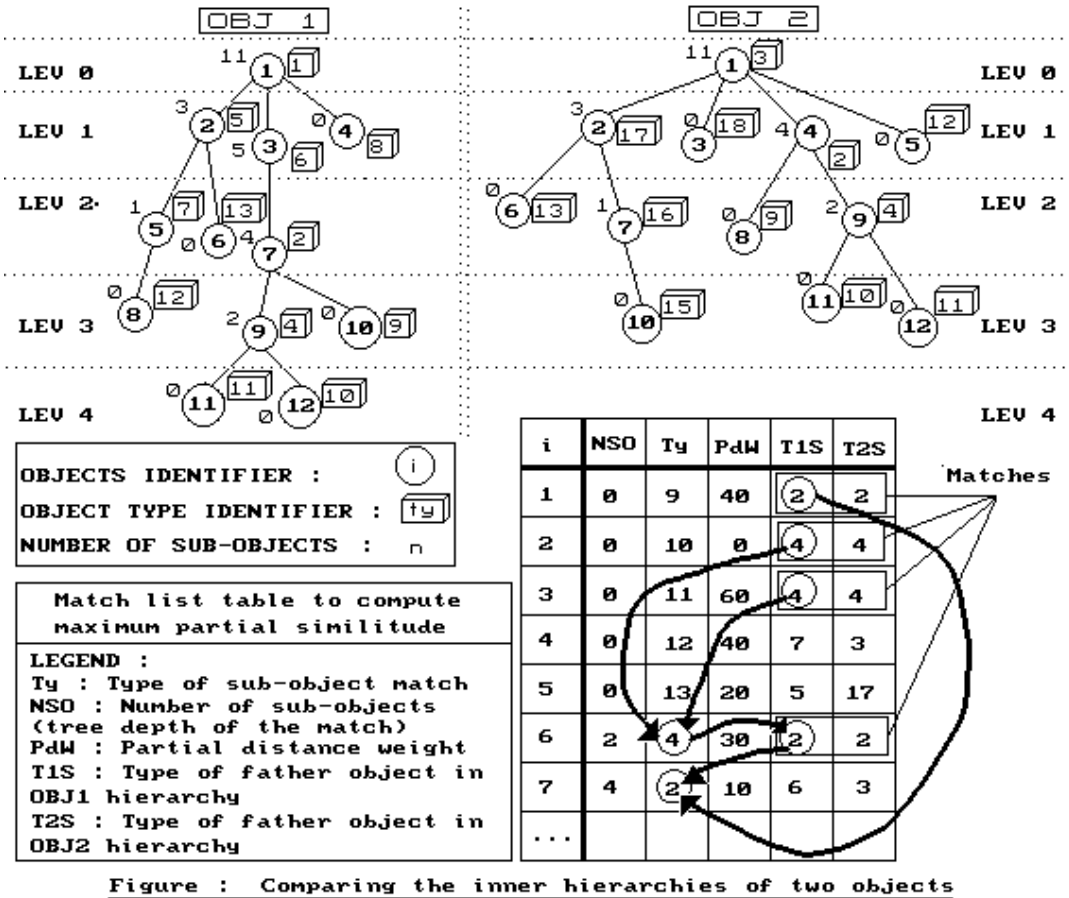
For each level of OBJ1 and OBJ2 structures, first the function builds two ordered lists of respective sub-object types, then it compares them and counts matches, taking into account the preponderance of some sub-objects by a weight established by database administrator. Matches are stored into a table which is dynamically built (see figure). The match table is exploited to find partial similarity involving sub-objects of OBJ1 and OBJ2 structure. The algorithm is implemented in C code.

It compares the inner composition hierarchies of two objects and tries to find as many sub-objects whose type match as possible.

The trees are examined depth first because :
- the hierarchies have generally not the same depth ;
- sub-objects of some complexity have to be compared.

CONCLUSION

We have proposed an O-O model which allows to solve some heterogeneities in a multi-database context. This model offers some qualities to be a good candidate as CDM in a federation because it offers semantic capabilities and characteristics for semantic relativism. This model authorizes the implementation of structure similarity algorithm by use of evaluation functions encapsulated in object application.
Currently, we define tools to implement value similarity.



i	NSO	Ty	PdW	T1S	T2S
1	0	9	40	2	2
2	0	10	0	4	4
3	0	11	60	4	4
4	0	12	40	7	3
5	0	13	20	5	17
6	2	4	30	2	2
7	4	2	10	6	3
...					

Figure : Comparing the inner hierarchies of two objects

FEW REFERENCES

(BOUL 90) BOULANGER D.

"*Heterogeneous Access in Database Management Systems*" Proceedings of IASTED International Symposium Applied Informatics, Innsbruck, Austria, February 1990.

(BOUL 92) BOULANGER D., COLLOC J.

"Cooperation between Heterogeneous Databases using an O-O Model" Working paper, 92/12, IAE, URA CNRS 1257, University of LYON 3, France, February 1992.

(BREI 90) BREITBART Y.

"Multidatabase Interoperability"

ACM SIGMOD Record, vol. 19, n°3, September 90, pp. 53-60.

(BROD 84) BRODIE M.

"On the Development of Data Models" in On Conceptual Modelling, Springer Verlag, TIS, 1984.

(COLL 87) COLLOC J., BOULANGER D.

"Conception d'un Système-Expert orienté objet destiné à l'aide au diagnostic médical" SITEF, Colloque Intelligence Artificielle et Santé, Toulouse, 1987.

(COLL 89) COLLOC J., BOULANGER D.

"Un modèle objet pour la représentation de connaissances empiriques"

Colloque International sur l'Informatique Cognitive des Organisations, ICO'89, Québec, Canada, juin 1989, pp. 119-140.

(COLL 90) COLLOC J.

*"Une approche orientée objet pour l'élaboration d'applications médicales"*Thèse de Sciences : Informatique et Automatique Appliquées, INSA de Lyon, n°0054 1990, 391 pages.

(DAYA 84) DAYAL U., HWANG H.

"View Definition and Generalization for Database Integration in a Multidatabase System" IEEE Transactions on Software Engineering, vol. SE-10, n° 6, November 1984.

(FU 83) FU K.S.

"A Step Towards Unification of Syntactic and Statistical Pattern Recognition"

IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-5, n° 2, March 1983, pp. 200-205

(JOHA 89) JOHANNESON P., WANGLER B.

"The Negotiation Mechanism in a Decentralized Autonomous Cooperating Information Systems Architecture"

IFIP 89, INFORMATION PROCESSING 89, Ritter G. (ed.), Elsevier Science Publications BV (North Holland).

(LITW 90) LITWIN W., MARK L., ROUSSOPOULOS N.

"Interoperability of Multiple Autonomous Databases"

ACM Computing Surveys, vol. 22, n° 3, September 1990.

(MANA 88) MANAGO M.

"Intégration de techniques numériques et symboliques en apprentissage automatique" Thèse de Sciences, Université Paris XI, n° 386, 1988, 195 pages.

(MOTR 87) MOTRO A.

"Superviews : Virtual Integration of Multiple Databases"

IEEE Transactions on Software Engineering, vol. SE-13, n° 7, July, pp. 785-798.

(SALT 91) SALTOR F., CASTELLANOS M., GARCIA SOLACO M.

"On Canonical Models for Federated DBs"

SIGM RECORD, volume 20, n° 4, december 1991, pp. 44-48.

(SHET 90) SHET A., LARSON J.

"Federated Database Systems for Managing Distributed Heterogeneous and Autonomous Databases" ACM Computing Surveys, vol. 22, n° 3, September 1990.