Resolving Semantic Heterogeneity in Databases with a Terminological Model: Correspondence Refinement

Ounissa Larab *, Aïcha-Nabila Benharkat **
Laboratoire d'Ingénierie des Systèmes d'Information
Institut National des Sciences Appliquées de Lyon
20 av, Albert Einstein, bât 502, F-69621 Villeurbanne CEDEX
* nissa@lisiecrin.insa-lyon.fr , **nabila@lisiecrin.insa-lyon.fr
Télécopie: 72.43.87.13 ☎:72.43.88.79

Abstract

The success of schema integration in multidatabase systems relies heavily on the determination of complete and refined correspondence relationships between them. So, the candidate schemas to be integrated must be rich and precise semantically, i.e., each of their data elements[1] must be sufficiently defined for to be distinguished from others or identified to some of them. Our schema integration method for federated databases is a terminological reasoning-based approach. It deals with the integration of terminologies that translate the export schemas (parts of DB[2] schemas which participate to the federation). In spite of it's semantic power, the terminological formalism can't express entirely and precisely the real world semantics of schema data elements .

To achieve this goal, we think that it is necessary to define refined and complete correspondence relationships between terms. Then, we propose to extend their semantics by a set of additional implicit knowledge which is not represented neither at the description level of terms nor at their extension level. It expresses semantic properties in connection with either local or global context of terminologies that participate to the federation.

1 Introduction and Motivation

Nowadays, communication between information systems is a challenge for most industrial firms. There are a lot of applications that require accessing and manipulating data from various pre-existing databases located in heterogeneous hardware and software environments anddistributed among nodes of computer network. Database integration seems to be the best solution to achieve this goal.

Systems dealing with distributed databases are generaly called *multidatabase systems.* We can distinguish two kinds : those relying on the schema integration mechanism and those relying on interdependant data, and transaction management mechanism [Rusinkiewicz andKarabatis, 1991].

Multidatabase systems using schema integration mechanism are our interest and especially those known to be loosely coupled systems (federated systems) where, only parts of database schemas (export schemas) [Benharkat and Larab, 1995] are integrated in one or several partial integrated schemas [Hsiao, 1992], [Sheth and Larson, 1990].

Semantic heterogeneity has always been a challenge for integration methodologies. To cope with this problem, we propose a method based on a terminological formalism where, all export schemas are translated in this formalism and then, all their data elements are compared to find their correspondence relationships. This comparison is based on the terminological system BACK [Hoppe et al., 1993] used as a helping tool because of its reasoning and classification power. Unfortunatly, in spite of all theses capabilities, it stays insufficient facing the semantic heteogeneity problem. Two descriptions declared to be equivalent by the BACK system may have different real world semantics. The contribution of our correspondence refinement method is to identify as musch as possible, the right semantic relationship between two data elements and to give the correspondingintegration rule.

Most of the methods based on terminological reasoning [Blanco et al., 1994] perform schema comparison by using the intensional level to find the structural correspondences and the extensional level to find the semantic ones. We think that the extensional level is not sufficient in the federated database context, because of the autonomy dimension of local DBMSs. There are no synchronous update operations between

[1] *Data elements: constructors and abstractions of any DB model (class, object, entity, relation, attribute, property...etc).*
[2] *DB: Database*

them. Therefore, two equivalent export schemas can have different extensions.

So, we only use the conceptual level, which is called the description level of terminologies. However, in considering the description level only to determine correpondence relationships, one could be faced to the *semantic relativism* problem [Spaccapietra and Parent, 1994]. This might happen either because the data model supports equivalent constructs or because designers have different perceptions ofthe same reality.

To solve this problem and particularly to be able to define refined and complete correspondence relationships between terms of terminologies (section 3), we propose to extend their semantics by a set of additional knowledge which is implicit [Larab and Benharkat, 1996a], i.e., it is represented neither at the description level of terms nor at their extension level. We designate such knowlegde by ***semantic properties*** which are added by means of a set of operators that we have defined. To find the correspondence relationships between sets of semantic properties of two terms, we use an ***heuristic*** that we have defined to compute a value: ***semantic measure***. The semantic measure helps us to find the right nature of semantic relationships that could exist between the semantic property sets.

In the following sections, we designate by *semantic enrichment phase* the adding step of the semantic properties which contribute to the refinement of correspondence relationships. Section 2 of this paper gives a short presentation of integration methods that use a CDM[3] . Section 3 gives an introduction of description logics. Section 4 explains how we perfom semantic enrichement of terms. Section 5 and 6 present the different types of semantics properties, the correspondence relations between them, and the correspondence relations between term descriptions. Section 7 presents the refinement process of correspondence relatonships and we conclude our work in section 8.

2 Related work on canonical data models

In the litterature, there exist many integration approaches relying on the translation of all schemas in a CDM, namely the entity-relationship model approach [Batini et al., 1986], [Grison, 1994], the object-oriented model approach [Thieme and Siebes, 1993], and the logical model approach [Bouzeghoub and Comyn-Wattiau, 1990], [Sheth et al., 1993], [Blanco et al., 1994]. In our case, we have chosen terminological logics as a canonical data model and the BACK system (KRS[4]) as a helping tool using its classification and inference possibilities.

3 Terminological Model

Description logics[5] [Nebel, 1990], were developed through research in artificial intelligence as tools for representation and reasoning. They describe object structures in terms of *concepts* and *roles*. Several systems implementing these logics are issued from KL-ONE [Brachman and Schmolz, 1985] namely, BACK, LOOM [Macgregor, 1991],...etc. *Taxonomic reasoning* is exploited by the terminological logics in many applications [Bergida, 1992], [Borgida, 1993], because of its ability to classify concepts in a taxonomy with the *subsumption* as a partial order relation. It also offers the advantage of performing a consistency check of class descriptions and their instance assertions.

The BACK formalism presents three characteristics: 1) it is based on logics, 2) it divides the knowledge into two components: *the terminological knowledge* called TBOX and *the assertional knowledge called* ABOX, 3) it includes a reasoning mechanism using the *subsumption* to organize terms in a taxonomy. The first basic notion of this language is the *concept* that represents a set of instances either intensionally or extensionally. Then the *role* represents a binary relation between concept instances. The *term* can either be a concept or a role. Finally, the *object* is an instance of one or more concepts.

Example of a terminology:
person:<anything,
registration :< domain(student) and range(string),
note :< domain(student) and range(number),
student := person and atleast(1, registration),
good_student := student and all(note,gt(16)).

4 Semantic enrichment

Before any integration operation, we allow the database administrator (DBA) to add semantic properties to terms by means of a set of operators. We have defined two kinds of operators for adding semantic properties: ***implicit property*** and ***relation property***. In the following, we use $\mathbf{T_i}$ to designate one terminology and $\mathbf{T_i.t_j}$ to designate one term of one terminology.

4.1 Implicit property

It expresses a simple knowledge which is absent from both the description and the extension of a term. It is related to the context of a terminology. It is represented by first order predicate **name_sp($\mathbf{T_i.t_j}$, sv).** Where,

[3] *CDM: Canonical Data Model*

[4] *KRS: Knowledge Representation System*

[5] *Also called Terminological logics*

name_sp represents the ***name*** of implicit semantic property and **sv** represents its ***value***.

Example:
T1.salary :< range(number). % number type %
T2.salary :< range(number). % number type %
The two "salary" terms have the same description (structure) but they have not the same semantics, because of their different implicit property values :
period(T1.salary, *'month'*). / *period*(T2.salary, *'week'*).
currency(T1.salary, *'franc'*)./ *currency*(T2.salary, *'dollar'*).

4.2 Relation property

It expresses relations between two or several terms of different terminologies. We represent it as the following first order predicate **relation-name (T_i.t_j,L).** Where, **relation-name** is a relation that exists between a term **T_i.t_j** and two or several terms in the liste **L**. This predicate is used to expess many relations related to the global context of integration, namely the relations between terms and those between their values. Corresponding operators are:

- **synonym(T_i.t_j,L_j).** and **homonym(T_i.t_j, L_j).** To express relationships between term names.
- **same_role(T_i.t_j,L_j).** To designate terms having same the role (e.g., *identifiers).*
- **frag_meaning(T_i.t_j,L_j).** To specify the general meaning **vs** special meanings of a term (e.g., *Parents* **vs** *father* and *mother*).
- **function(T_i.t_j,L_j).** To specify the corresponding conversion function between two terms (e.g., *temperature* term can be in *Celsius degree* in one terminology and in *Fahrenheit* degree in another).
- **val_corresp(T_i.t_j,L_j, T_q.t_k,L_k).** To set that the same term can have different value types in different terminologies (e.g., *note* term can have string values in one terminology and number values in another).

In the following sections, we use the notation <cor_o> to designate these semantic relation properties.

5 Correspondences between implicit property sets : <cor_p>

Let us consider, two sets of implicit properties P and P' added to terms **T1.t_i** and **T2.t_j**. where **P** has a cardinality α and **P'** has a cardinalty β, α ≤ β. We define a test **[$S_{k=1,\alpha}$ (p_{ik})P']** to look for a property in P' that matches a property (p_{ik}) ∈ P (figure 1).

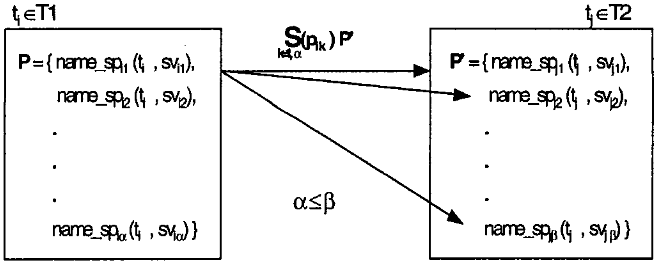

t_i∈T1
t_j∈T2
$S_{k=1,\alpha}(p_{ik})$ P'
P={ name_sp_{i1} (t_i , sv_{i1}),
name_sp_{i2} (t_i , sv_{i2}),
name_$sp_{i\alpha}$ (t_i , $sv_{i\alpha}$) }
α≤β
P'={ name_sp_{j1} (t_j , sv_{j1}),
name_sp_{j2} (t_j , sv_{j2}),
name_$sp_{j\beta}$ (t_j , $sv_{j\beta}$) }

Figure 1: Matching test of P with P'

The matching test is repeated for each property in P. Its result is a value **r_i = 1** or **r_i = 0.1** or **r_i = 0**. Each value represents one of the following situations:

- **r_i = 1** : Property in P has the same name and the same value than one in P':
  name_sp (T1.t_i, *sv*)∈P / ***name_sp*** (T2.t_j, *sv*)∈P'
- **r_i = 0.1** : property in P has the same name but a disjoint value than one in P' :
  name_sp (T1.t_i, sv_1)∈P / ***name_sp*** (T2.t_j, sv_2)∈P'
- **r_i = 0** : property in P has neither the same name nor the same value than one in P' :
  name_sp₁ (T1.t_i, sv_1)∈P / ***name_sp₂*** (T2.t_j, sv_2)∈P'

All the answers (r_i) of a matching test **S** are grouped in an **answer-vector R [r_1, r_2,... r_α].**

Now, we determine the correspondence relationship **<cor_p>** between P and P'. We first, compute the semantic measure **'M_s'** using an **heuristic** that we have defined, it is a simple calculus function :

$$\boxed{M_s = \sum_{i=1,\alpha} r_i \;/ r_i \in R}$$

and according to the value **M_s** we determine **<cor_p>(P,P') = φ (M_s)** as in figure 2.

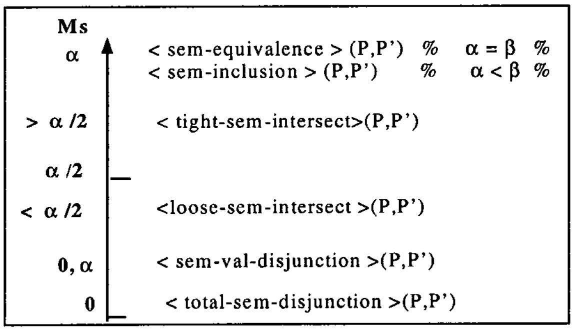

Ms
α
< sem-equivalence > (P,P') % α = β %
< sem-inclusion > (P,P') % α < β %
> α /2
< tight-sem-intersect>(P,P')
α /2
< α /2
<loose-sem-intersect >(P,P')
0, α
< sem-val-disjunction >(P,P')
0
< total-sem-disjunction >(P,P')

Figure 2 : Semantic taxonomy of <cor_p>(P,P').

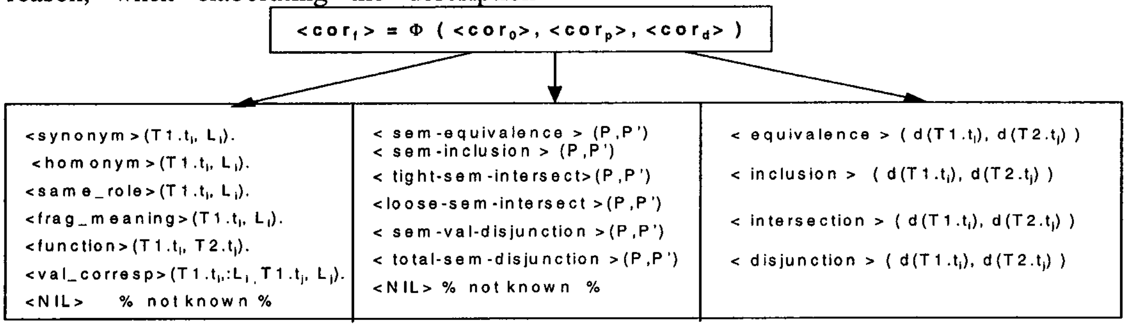
6 Correspondence between term descriptions : <cor_d>

To determine <cor_d> between two term descriptions, we use the BACK system as a helping tool because of its reasoning and classification power. All <cor_d> obtained automatically by the BACK system are those referring to usual set relationships (equivalence (≡), inclusion (⊆), intersection (∩), and disjunction (≠)).

So, we note that, the BACK system considers neither the schematic conflicts nor the the semantic ones that may exist between related terms. It allows us to find coherent correspondence relationships but, even though its reasoning power, it has some failings when structural or semantic conflicts occur between terms. For this reason, when elaborating the description correspondence relationship process, we consider all these failings *'case by case'.* [Larab, 1996].

7 Fine correspondence relationships : < cor$_f$>

Fine (refined) correspondence relationships are obtained according to the combination of all correspondences determinedabove as shown in figure 3:



<cor$_f$> = Φ (<cor$_0$>, <cor$_p$>, <cor$_d$>)
<synonym>(T1.t$_i$, L$_i$).
<homonym>(T1.t$_i$, L$_i$).
<same_role>(T1.t$_i$, L$_i$).
<frag_meaning>(T1.t$_i$, L$_i$).
<function>(T1.t$_i$, T2.t$_j$).
<val_corresp>(T1.t$_i$,:L$_i$ T1.t$_j$, L$_j$).
<NIL> % not known %
< sem-equivalence > (P,P')
< sem-inclusion > (P,P')
< tight-sem-intersect>(P,P')
<loose-sem-intersect >(P,P')
< sem-val-disjunction >(P,P')
< total-sem-disjunction >(P,P')
<NIL> % not known %
< equivalence > (d(T1.t$_i$), d(T2.t$_j$))
< inclusion > (d(T1.t$_i$), d(T2.t$_j$))
< intersection > (d(T1.t$_i$), d(T2.t$_j$))
< disjunction > (d(T1.t$_i$), d(T2.t$_j$))

Figure 3 : Refinementprocess of correspondencerelationships

Example 1 :

t1 = T1.employee :< person and exactly(1,id)
and exactly(1,dept) and atleast (1,adress).
P = {*activit_typ*(T1.employee, *'public'*),
activity_dom(T1.employee, *'education'*)}

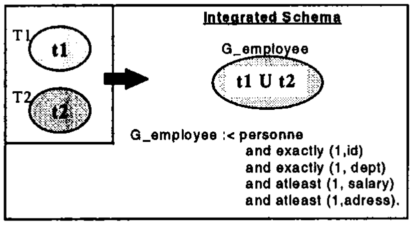
t2 = T2.worker :< person and exactly(1, id)
and atleast (1, salary) and exactly(1, dept).
P' ={ *activit_typ*(T2.worker, *'public'*),
activity-dom(T2.worker, *'education'*) }

-***Correspondances between terms*:**
<cor$_0$> = <synonym>(T1.employee ,T2.worker) ,
<cor$_p$> = < sem-equivalence >(P,P') ,
<cor$_d$> = < ∩> (d(T1.employee), d(T2.worker))

- ***Fine correspondence between terms*:**
< **corf** > = Φ (<synonym >, <sem-equivalence>, < ∩>)
= < equivalence> (T1.employee ,T2.worker)

- ***Integration rule*:**
Semi-automatic choice (because of the synonymy) of the name of the global term and automatic union of the two term descriptions.

- ***Integrated schema*:** % Global employee term %
G_employee = (T1.employee ∪ T2.worker)
G_employee:< person and exactly(1,id) and exactly(1, dept) and atleast(1, salary) and atleast(1, adress).



Example 2 :

t = T1.salary :< range (integer). % number type %
P={*period*(T1.salary,'*month*'), *currency*(T1.salary, '*franc*')}

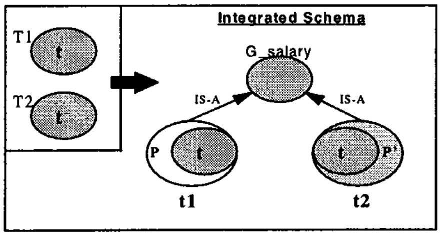
t = T2.salary :< range (integer). % number type %
P'={*period*(T2.salary,'*week*'), *currency*(T2.salary,'*dollar*')}.

- ***Correspondances between terms*:**
<cor$_0$> = <NIL> % unknown relation %
<cor$_p$> = < sem-val-disjunction> (P,P')
<cor$_d$ > = < ≡> (d(T1.salary),d(T2.salary))

- ***Fine correspondance between terms*:**
<**cor$_f$** > = Φ (<NIL>, < sem-val-disjunction >, <≡>)
= < exclusive-generalization >(T1.salary ,T2.salary)
It expresses that the two terms have a common general semantics but not a common special semantics.

- ***Integration rule:***
Automatic creation of a global term with a general semantics G_d(T1.salary) and two special terms :
t1= d(T1.salary) and P , t2= d(T2.salary and P') .

- ***Integrated schema*:**
t = G_salary :< range (integer). % Global salary term %
t1 = T1.salary :< G_salary and period : 'month'
and currency : 'franc'.
t2 = T2.salary :< G_salary and period : 'week'
and currency : 'dollar'.



8 Conclusion

In this paper we present the advantage of using a Terminological KRS to find most of the structural correspondence relationships between translated data elements. Furthermore, it allows us to check their coherence and their correctness. However, the correspondences at the semantic level can't be obtained correctly and entirely. So, we have defined a set of operators to add a set of semantic properties to terms and an heuristic that helps finding the right semantic links between terms.

The *conjunction* of the terminological reasoning and the semantic property concept contributes to the refinement of the correspondence relationships between terms, i.e., it permits to express precise semantic link that could exist between two terms, and to make easier their semi-automatic determination.

Our actual work is essentially the definition of precise semantics of each refined correspondence relationship and, for each refined correspondence relationship we also define, the corresponding conflict resolution and integration rules.

References

[Batini et al., 1986]C. Batini, M. Lenzerini, S.B. Navathe. A Comparative Analysis of Methodologies for Database Schema Integration. *ACM Computing surveys*, pages 323-364 ,Vol 18, No 4, December 1986.

[Benharkat and Larab, 1995] A. Nabila Benharkat, Ounissa Larab. Using a Knowledge-Based Representation System in Database Integration. *Eighth International Conference on Software Engineering & Its Application*, pages 453-468,France, November 1995

[Bergamashi and Sartori, 1992] S. Bergamachi, C. Sartori, On Taxonomic Reasoning in Conceptual Design. *ACM transactions on database systems*, pages 385-422, Vol 17, No 3, September 1992.

[Blanco et al., 1994]J.M. Blanco, A. Illaramandi, A. Gono, J.M. Pérez, Using a Termonological System to Integrate Relational Databases. *Information Sysems Design and Hypermedia, CEPADUS-EDITIONS*, 1994.

[Borgida, 1992] A. Borgida. Description Logics are not just for the Flightless Birds: A new look at the utility and Foundations of Description Logics. *Technichal Report*, Dept of Computer sciences, Rutgers University, June 1992.

[Borgida, 1993] A. Borgida. Loading Data into Description Reasoners. In *Proc. of the ACM SIGMOD, International Conference Managment Data, SIGMOD RECORD*, pages 217-226, Washington DC, June 1993.

[Bouzeghoub and Comyn-Wattiau, 1990] M. Bouzeghoub, I. Comyn-Wattiau.View Integration by Semantic Unification and Transformation of Data Structures. In *Proc. of the IEEE 6th Int. Conf. Entity Relationship Approach*, Lausanne, October 1990.

[Brachman and Schmolz, 1985] R.J. Brachman, J.G. Schmolz. An overview of KL-ONE knowledge representation system. *Cognitive Science*, 9(2), pages 171-216, April 1985.

[Grison, 1994] T. Grison. Integration de Schémas de Bases de Données Entité/Association. *Thèse de Doctorat de l'Université de Bourgogne*, France, 217 pages, 1994.

[Hoppe et al., 1993] T. Hoppe, C. Kinderman, J.J. Quani, A. Schniedel, M. Fisher. BACK V.5: Tutorial & manual. *Projekt KIT-BACK*, Technishe Universität Berlin, Institut Für Software und Theoretishe Informatik, Berlin Germany, 1993.

[Hsiao, 1992] D.K. Hsiao. Federated databases and systems : Part I - A tutorial on their data sharing. *VLDB Journal*, pages 127-179, 1, 1992.

[Larab and Benharkat, 1996a] O. Larab, A.N. Benharkat. Propriétés sémantiques et Raffinement des Relations de Correspondances dans l'Intégration des Schémas Hétérogènes basés sur le Modèle Terminologique. *XIVe CONGRES INFORSID*, pages 179-197, Bordeaux France, juin 1996.

[Larab, 1996] O. Larab. Terminological Reasoning-Based Approach of Correspondence Refinement in Multidatabase Systems. *Research Report RR-96-7,* 16 pages, LISI INSA-Lyon 1996.

[Macgregor, 1991] R. Macgregor. Inside the LOOM description classifier. *SIGART Bulletin*, 2(9), pages 88-92, 1991.

[Nebel, 1990] B. Nebel. Terminological Reasoning and Revision in Hybrid Representation Systems. *Lecture Notes in Artificial Intelligence, Springer Verlag*, 267 pages, 1990.

[Rusinkiewicz et al., 1991] M. Rusinkiewicz, A. Sheth, G. Karabatis. Specifying Interdatabase Dependencies in a Multidatabase Environment. *IEEE Computer*, pages 46-53, 24(12), December 1991.

[Sheth and Larson, 90] A.P. Sheth, J. Larson. Federated database systems for managing distributed, heterogeneous and autonomous databases. *ACM Computer surveys*, pages 183-236, Vol.22, N°3, September 1990.

[Sheth et al., 1993] A.P. Sheth, S.K. Gala, S.B. Navathe. On automatic reasioning for schema integration. *International Journal of Intelligent and Cooperative Information Systems*, pages 23-50, Vol. 2, N°. 1, April 1993.

[Spaccapietra and Parent, 94] S. Spaccapietra, C. Parent. View Integration: A Step Forward in Solving Structural Conflicts. *IEEE Transactions on Knowledge and Data Engineering*, pages 258-274,Vol. 6, N°. 2, April 1994.

[Thieme and Siebes, 1993] C. Thieme, A. Siebes. Schema Integration in Object Oriented Databases. In *CAISE'93, 5th International Conference, Springer Verlag*, pages 54-70, 1993.