FORMAL FRAMEWORK FOR SEMANTIC INTEROPERABILITY

Nadia Yaacoubi Ayadi, Mohamed Ben Ahmed
RIADI-ENSI
Campus Universitaire, 2010 La Manouba

Yann Pollet
Chaire d'intgration des systmes, CNAM
292, Rue Saint Martin

Keywords: Semantic Interoperability, Logical Formalization, Mapping Approaches, Ontology.

Abstract: Semantics of schema models is not explicit but always hidden in their structures and labels. To obtain semantic interoperability we need to make their semantics explicit by taking into account both the interpretation of the labels and the structures described by the arcs. We address in this paper the issue of semantic interoperability between systems relying on semantically heterogeneous hierarchies, having been designed for the purpose of independent specific goals and activities. Given a set of generalization hierarchies, our approach gives much emphasis on *semantics added-value* by "emerging" the intended informal meaning of concepts, we rely on Wordnet lexical repository. In the first part of the paper, we provide a rigorous logical framework for representing and automatically reasoning on generalization hierarchies except their formalism (UML, ER diagram, etc). Then, we describe The SEM-INTEROP algorithm that consists on two main steps : *semantic interpretation* and *semantic comparaison*.

1 INTRODUCTION

Knowledge sharing between heterogeneous sources is a significant challenge, which has been the focus of much research but remains an open problem. Enabling the cooperation of heterogeneous information systems is not easy to achieve because related knowledge is disparate and described in different terms and using different assumptions. Heterogeneity may arise from syntactic, structural and semantic discrepancies in information systems. *Syntactic heterogeneity* is due to the use of diverse database models (object-oriented vs relational), *structural heterogeneity* arises from different conceptual choices during the conceptualization phase (modelling as a class, as a relationship, or as an attribute), and *semantic heterogeneity* comes from differences between the terms used to represent information and their intended meaning (Kashyap and Sheth, 1996).
In this paper, we focus on semantic heterogeneity and interoperability solutions that address this aspect of semantic heterogeneity. Of course, the presence of a variety of conceptual models is unavoidable both because humans think differently and because the applications of these models were designed for different needs. Thus, the fundamental question in any approach to interoperability of information systems is that of identifying concepts or a set of concepts in different information systems that are semantically related, and then resolving the schematic differences among semantically related concepts (Sheth and Kashyap, 1993). By schematic differences, we may refer to different partial representations of a same concept, different granularity-level description or a perspective representation when it encodes a spatio-temporal, logical, and cognitive point of view.

Two main categories of frameworks have been proposed for the co-operative information systems : *federation of information systems* (Sheth and Larson, 1990) and *mediation* (Wiederhold, 1992; Chawathe et al., 1994) which relies on the definition of *wrappers* and *mediators*. Mediation-based architectures facilitate evolution through the addition of new data sources. They support cooperation of large information systems and thus are more suitable in web environment. Federation-based architectures are best suited for small-scale cooperation.
Irrespectively of the system architecture, a fundamental task in integration is the ability to recognize an a-priori agreement on knowledge shared by communities through describing mappings between them and supporting access to the existing data instances.

Yaacoubi Ayadi N., Pollet Y. and Ben Ahmed M. (2006).
FORMAL FRAMEWORK FOR SEMANTIC INTEROPERABILITY.
In *Proceedings of the First International Conference on Software and Data Technologies*, pages 139-144
DOI: 10.5220/0001319301390144

A large number of papers have investigated various facets of mapping, such as mapping discovery, mapping definition or mappings usage (for a survey see (Rahm and Bernstein, 2001)).

In such a distributed setting, we believe that an *a-priori* agreement on knowledge and knowledge exchange is very hard to achieve. Indeed, if we try to achieve integration or interoperation of large and disparate information systems, the current standard approach of creating large-scale shared knowledge will hardly scale up to the size of the (semantic) Web, and is also conceptually problematic because in our opinion *knowledge is never context-free* (Yaacoubi and BenAhmed, 2003; Stoœmer and Stecher, 2005), and can thus never be perfectly shared.

In this work, our objective is to propose a complete approach for the semantic integration of Generalization Hierarchies. We adapt previous results on schema and ontology integration (ontology fusion, ontology mapping, ontology alignement for a survey, see (Wache et al., 2001)) to tackle different kinds of heterogeneities one might encounter during the interoperation of information systems. Indeed, we think that the semantics of schema models is not explicit but is hidden in their structures and label's concepts. Given a set of generalization hierarchies, our approach gives much emphasis on *semantics added-value* by "emerging" the intended informal meaning of their concepts through mapping them to Wordnet[1] ontology, but also through interpreting their structural position.

The aim of this paper is to describe an algorithm to analyse the implicit knowledge in order to provide correct mappings between concepts. First, we propose a logical formalization of class hierarchies. Thus, we provide a rigorous logical framework for representing and automatically reasoning on generalization hierarchies except their formalism (UML, ER diagram, etc). The SEM-INTEROP algorithm performs two main steps : *semantic interpretation* and *semantic comparaison*.

Compared to other related works, our proposal falls within the scope of approaches that aim at defining a formalism or methodology to specify and use inter-schema correspondences. We can assume that an initial set of inter-schema correspondences given by the designer, however we don't consider the subject of query reformulation, which is out of the scope of this paper. The proposal contributes to the area of research on the following original topic :

- A semantic interpretation approach combining linguistic, structural and contextual knowledge is proposed in order to be able compare semantically concept's hierarchies,
- We propose a *mapping algebra* that can be intressent to realize schema transformations.

The paper is structured as follows : Section 2 presents logical constructs for generalization hierarchies. In section 3, we present our semantic-based approach for interoperability, we describe the first version of the SEM-INTEROP algorithm. Finally, Section 4 concludes the paper and identifies future works.

2 BASICS OF THE APPROACH

Let us first clarify our terminology. In the litterature, we identify four levels of abstractions. At the bottom level we have actual *data* (or *instances*) organized according to a variety of (semi) structured formats (relational tables, XML documents, HTML files, scientific data, and so on). At the second level we have *schemes*, which describe the structure of instances (a relational schema, a DTD, an XML schema or one of its dialects, etc.). Then, we have different formalisms for the description of schemes that we call *models* (e.g. conceptual model like the ER model or UML class diagram). Finally, we use the term *metamodel* to mean a general formalism for the definition of various models. Specifically, a *metamodel* is made of a set of *metaprimitives*. Each metaprimitive captures a class of constructs of different data models that share a common characteristics or, more precisely, that implement, possibly with different names, the same basic abstraction principle (Torlone and Atzeni, 2001). Examples of metaprimitives : class, attribute, definition domain, relationship, generalization, disjoint union, key, foreign key, and so on.

Here, we introduce more specifically and formally the terms of our problem. As conceptual model, we opt for Generalization– its inverse: *specialization*– Hierarchies. We propose a logical formalism that allows us to uniformly represent heterogeneous hierarchies.

Definition 1 (Generalization hierarchy) *We define a class hierarchy $\mathcal{H}$ as a triple $\langle \mathcal{C}, \mathcal{E}, \Phi \rangle$:*

- *$\mathcal{C}$ is a finite set of classes, $\mathcal{C}=\{c_i\}$, each class c_i is characterized by a name and a set of attributes, $c_i=\langle n_{c_i}, \mathrm{A}(c_i) \rangle$. Each attribute $a_h \in \mathrm{A}(c_i)$, with* h=1,...,n *is defined as a pair, $a_h = \langle n_h, d_h \rangle$, where n_h is a name and d_h is the domain associated with a_h, respectively.*
- *$\mathcal{E}$ is a set of arcs on $\mathcal{C}$, for instance, $\mathcal{E}$ is a set of subsumption relationships (ISA relationships) between classes.*

[1] Wordnet is available at http://wordnet.princeton.edu.

- Φ *is a logical interpretation of* $\mathcal{H}$ *which make explicit all the knowledge like attributes, values domains or constraints embedded in* $\mathcal{H}$ *by means of a consistent logical formulation. We assign to each parent class an AND/OR logical formulae expressing constraints among instances of child classes.*

Informally, one can use a *generalization* between two classes to specify that each instance of subclass is also an instance of the superclass. Hence, instances of the subclass inherit the properties of the superclass, but typically they satisfy additional properties that in general do not hold for the superclass. Figure 1 shows a generalization hierarchy example represented with the Unified Modelling Language (UML) constructs [2].

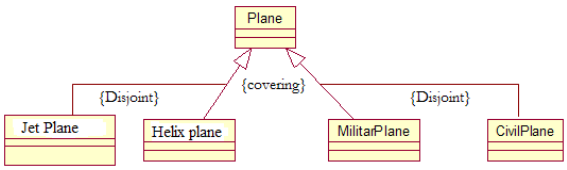

Figure 1: Example of an UML generalization hierarchy.

In our approach, a class C generalizing a class C_1 can be captured by means of the following logical assertion :

$$\textbf{ISA}(C_1,C) \Rightarrow \forall\, x,\, C_1(x) \subset C(x)$$

With regard to generalization hierarchy, semantic constraints related to the intersection of the sibling classes– that is, classes having a common superclass –are often proposed, allowing the notions of disjoint and completeness constraints to be introduced. In particular, a generalization is disjoint or overlapping depending on whether the intersection of the siblings classes is empty or not, respectively. These constraints may be captured by means of the following logical assertions:

$$\textbf{ISA-ASSERT}(C,[Constraint])$$

Disjointness constraint among C_1, C_2,..., C_n can be expressed by the following predicate and assigned to the superclass C :

$$\forall \text{ i=1,...,n }, C_1 \textbf{ XOR } C_2 ... \textbf{XOR } C_n \Rightarrow \forall\, x,\, C_i(x) \supset \bigwedge_{j\in\{1..n\}\setminus i}^{n} \neg\, C_j(\mathrm{x})$$

The *complete* constraint expressing that each instance of C is at least one of C_1, ...,C_n is expressed by :

$$\forall\, x,\, C(x) \supset \bigvee_{j=1}^{n} C_j(\mathrm{x})$$

Referring to figure 1, specific constraints hierarchy can be captured by means of logical expressions:

[2]see the last specification of UML on http://www.uml.org

Example 2 ***ISA*** *(Plane, JetPlane)*
ISA *(Plane, HelixPlane)*
ISA *(Plane, CivilPlane)*
ISA *(Plane, militaryPlane*
ISA-ASSERT *(Plane,* $\{$*Jetplane* ***XOR*** *Helixplane*$\}$ $\bigcup$ $\{$*Civilplane* ***XOR*** *militarplane*$\}$*)*

Example 3 *Referring to figure 2, we can define the following predicates considering* C_{13} *as a subclass of* C_1 *and* C_3 *(respectively to* C_{24}*):*
ISA*(*C_{13}*,* C_1*)*
ISA*(*C_{13}*,* C_3*)*
ISA-ASSERT*(*C_{13}, $C_{13} \subset C_1 \bigwedge C_3$*)*

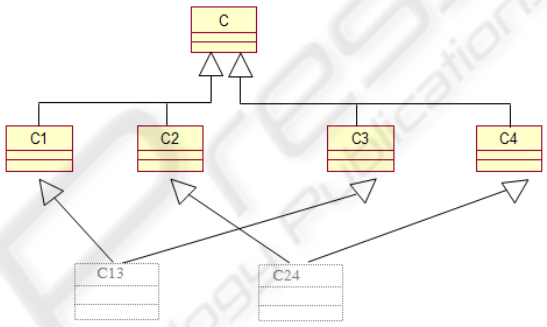

Figure 2: Multiple level hierarchy.

Disjointness and complete constraints are in practice the mostly commonly used constraints in generalization hierarchies. Finally, we may express additional constraints specifying for example restrictions on domain values.

The *logical formulation* of generalization hierarchies allows us to go far beyond. However, this logical formulation must be *consistent*.

Consistency of generalization hierarchies. Generalization hierarchies is *consistent*, if its classes can be populated without violating any of the constraints. By exploiting this logical formalization, the consistency of the hierarchy can be checked by checking the satisfiability of the corresponding knowledge base (logical assertions).

Class subsumption. A class C_1 is *subsumed* by a class C_2 if, whenever the constraints imposed by the generalization hierarchy are satisfied, the extension of C_1 is a subset of the extension of C_2. Such a subsumption allows one to deduce that properties for C_1 hold also for C_2.

Class equivalence. Two classes are *equivalent* if they denote the same set of instances whenever the constraints imposed by the generalization hierarchy

are satisfied. Determining equivalence of two classes allows for their merging.

In the next section, we describe our interoperalization approach that is based on logical formalization and also on linguistic and contextual knowledge.

3 INTEROPERALISATION APPROACH

We have seen in the previous section how a logical formulation can be associated to a given hierarchy $\mathcal{H}$ based on constraints expressed in conceptual models. Indeed, any model has no meaning in isolation. *Only through a semantic space (e.g. domain ontology) are its elements are linked to context, language, situation, actor, role, etc*. [3] The semantic space represents knowledge on a domain, while each model asserts a single proposition related to a specific context.
Commonly with (Bouquet et al., 2004), we can identify at least three distinct levels of knowledge which can used to *elicit* a schema's semantics:

- *Lexical knowledge.* knowledge about the meaning of words used to label classes and attributes. Indeed, word senses can be automatically generated from a Lexical Knowledge Base (LKB). Wordnet (Fellbaum, 1998) has been adopted in the current work because it is the largest repository of word senses and semantic relations currently available. However, Wordnet could be replaced by another combination of a linguistic resource and a domain knowledge resource.
- *Structural knowledge.* Knowledge deriving from the arrangement of classes in the generalization hierarchy. Instead, our analysis consider the *implicit information* deriving from the structural relations with other concepts of the hierarchy.
- *Domain knowledge.* Knowledge describing the logical structure of a specific domain, its concepts and the relations between them. For instance, Wordnet assigns a domain label (e.g., *tourism*, *zoology*, *sport*, etc.) to most synsets.[4]

In the current version of the algorithm, SEMINTEROP takes two generalization hierachies $\mathcal{H}_1$ and $\mathcal{H}_2$ as input and returns mappings between their structures. The algorithm performs the following main steps: *Semantic Interpretation* and *Semantic Comparison*.

3.1 Semantic Interpretation

In this phase, we make explicit the meaning of each class based on a linguistic interpretation. Compared with other approaches to schema matching such as (Madhavan et al., 2001; Bergamaschi et al., 1999), we do not limit ourselves to a linguistic analysis of labels. Instead, we extend this analysis by considering the *implicit knowledge* deriving from the *context* where the class appears. Then, we interpret constraints like *Disjointness*, *Covering*, *negation* in order to exhibit new abstractions of classes.

Linguistic Interpretation. Let $\mathcal{H}$ be a generalization hierarchy, and $\mathcal{C}$ are classes occurring in $\mathcal{H}$. Each class $c_i \in \mathcal{H}$ are described by labels, which in turn are composed by words and, possibly, separators between them. We define the lexicon of a given hierarchy $\mathcal{H}$ as $\mathcal{L}$= $\{l_1, l_2,...,l_n\}$ be a valid set of labels belonging to an hierarchy $\mathcal{H}$. The process of interpretation associates the appropriate WordNet synset S_k^i to each label l_k in $\mathcal{L}$. So, the sense of $\mathcal{L}$ is defined as:

$$\mathrm{S}(\mathcal{L})=\{\ S_k^i \parallel S_k^i \in \mathit{Synset}(l_k),\ l_k \in \mathcal{L}\ \}$$

where *Synset*(l_k) is the set of senses provided by WordNet for a label l_k. For instance, S(*Plane*)= {{*Airplane*#1}, {*Sheet*#2}, {*stage*#3}, {*planing machine*#4}, {*Carpenter's plane*#5}}.

Contextualization. Contexts appear in many disciplines as meta-informations to characterize the specific situation of an entity, to describe a group of conceptual entities, and to partition a knowledge base into manageable sets or as a set of logical constructs to facilitate reasoning services (Dey and Abowd, 1999). In the current work, we make use of the following meta-level properties (Guarino, 1998): **TYPE**, for synsets representing rigid properties e.g. a *person*, **ROLE**, for synsets representing anti-rigid properties e.g. *student*, and **ATTRIBUTION**, for synsets representing possible values of attributes e.g. *employee*, as an attribute-value for *activity*. These semantic constructs allow us to express *Contextualized Concepts* considering their structural and contextual features in terms of logical assertions.

Example 4 An *employee* is a person who has a role of a worker and has necessary a salary.

Employee(x) := Person(x) $\sqcap$ *(*$\exists$ *Role(x).worker)* $\sqcap$ *(*$\exists$ *Attribution(x).Salary)* $\sqcap$ *(*$\neg$ *Employer(x))*

A *student* is a person who has a role of a learner and is enrolled in one level.

[3] Adapted from Sowa, *"a conceptual graph has no meaning in isolation. Only through the semantic network are its concepts and relations linked to context, language, emotion, and perception*".

[4] Wordnet 2.0 also provides domain labels. However, we preferred the lable data set described in (Magnini and Cavaglia, 2000)

Student(x) := Person(x) $\sqcap$ *(*$\exists$ *Role(x).learner)* $\sqcap$ *(*$\exists$ *Attribution(x).level)* $\sqcap$ *(*$\exists$ *EnrolledIn(x).level)*

An *Adult Citizen* is a person who take an active role and he is an adult person.

Adult Citizen(x) := Person(x) $\sqcap$ *(*$\exists$ *Role(x).Activity)* $\sqcap$ *(*$\exists$ *Attribution(x).Adult)* $\sqcap$ *(*$\neg$ *Attribution(x).Juvenile)*

Implicit Constraints Interpretation. Implicit structural constraints can lead to derive new classes. For instance, covering constraint is interpreted as a *meet* operation among classes ($\downarrow$), the resulting class represents the greatest common lower bound, possibly equal to $\perp$– the least element in the hierarchy. We may obtain a semi-lattice as illustrated in figure 3, considering:

- ISA relationship as a partial order relation that is a reflexive, antisymmetric and transitive relation,
- Existence for each pair of classes a greatest common lower bound.

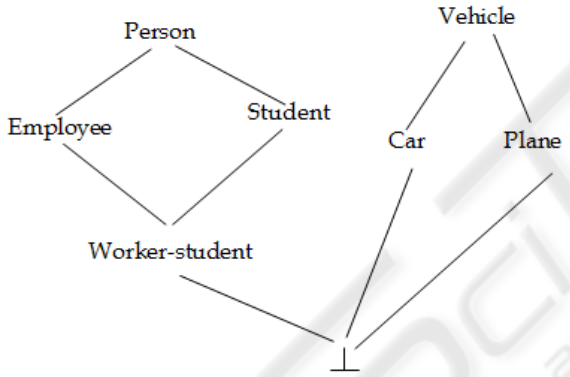

Figure 3: A semi-Lattice structure.

Conjunction between classes may be expressed in a logical formulae, for instance : *Worker-student := Employee* $\sqcap$ *Student*

3.2 Semantic Comparison

Intuitively, the problem of semantic interoperability arises when one needs to find relations between classes belonging to distinct (and thus typically heterogeneous) hierarchies. Formally, we define the problem of semantic interoperability as the problem of discovering mappings between classes in two distinct hierarchies $\mathcal{H}$ and $\mathcal{H}'$:

Definition 5 ***(Mapping)*** *A mapping* $\mathcal{M}$ *from* $\mathcal{H} = \langle \mathcal{C}, \mathcal{E}, \Phi \rangle$ *to* $\mathcal{H}' = \langle \mathcal{C}', \mathcal{E}', \Phi' \rangle$ *is a function* $M: \mathcal{C} \times \mathcal{C}' \longrightarrow \mathcal{R}$*, where* $\mathcal{R}$ *is the set the possible relations.*

We may distinguish two forms of mappings : *classical mapping* and *rule-based mapping*. The first form is widely used to express semantic relations between classes that are *equivalence mapping*, *disjointness mapping*.

Example 6 ***(Classical Mappings)***
Voiture(x) $\Rightarrow$ *Car(x)*
Car(x) $\Rightarrow$ *Voiture(x)*
Male(x) $\Rightarrow$ $\neg$ *Female(x)*

A rule-based mapping can be used to represent complex mappings such as generalization/specialization mappings.

Example 7 ***(Rule-based Mapping)***
CarOwner(x) $\Rightarrow$ *Person(x)* $\sqcap$ *(Attribution(x).Car)* $\sqcap$ *(Role(x).Owner)*

Mapping Algebra. Unfortunately, a few number of research works propose mathematical foundations for the mapping problem. Mapping classes belonging to different hierarchies is important but not sufficient. Depending on these mappings, how we can restructure internal organisation of given hierarchies to obtain the "*interoperation structure*" that represent their greatest common lower bound. For example, Considering hierarchies as a partially ordered sets, they can be considered to be equivalent, if there exists a *bijective* function between these sets which does also preserve the order (i.e. which is *monotonic*). In this case, being monotonic means that a function respects the internal structure of partially ordered sets, while bijectivity indicates the equivalence of two ordered sets. Structure-preserving functions are a typical implementation of what is called a *morphism*.

Two partially ordered sets $\mathcal{H}$ and $\mathcal{H}'$ are *equivalent* or *isomorph* whenever there is a monotone function $f : \mathcal{H} \rightarrow \mathcal{H}'$ that has a monotone inverse, i.e. for which there is a monotone function $g : \mathcal{H} \rightarrow \mathcal{H}'$ with $g \circ f = id_{\mathcal{H}'}$ and $f \circ g = id_{\mathcal{H}}$. We call a morphism an isomorphism if it has a (necessarily unique) inverse morphism.

For thus, We may develop a mapping algebra including operators such as: $\mathcal{S}$*-join* (Semantic Join), $\mathcal{S}$*-meet* (Semantic meet), $\mathcal{S}$*-Project* (Semantic Projection).

4 CONCLUSION AND FUTURE WORK

In this paper, we have provided a formal semantics for generalization hierarchies and then used that formal framework to explore a number of linguistic and semantic issues crucial for interpreting the knowledge

implicitly represented in such hierarchies. The algorithm we have proposed performs a linguistic interpretation of the labels provided in the hierarchy, based on the Wordnet Ontology. The process of interpreting labels is extended with a contextualization process which is a progressive construction of logical expressions where predicates constructs are based on three meta-properties : TYPE, ROLE and ATTRIBUTION.
Next, we perform a semantic comparison that consists on discovering mappings between classes. Besides classical mappings, we introduce rule-based mappings that express constrained complex mappings. We think that mapping two hierarchies $\mathcal{H}$ and $\mathcal{H}'$ means, at least, finding an isomorphic sub-hierarchy of $\mathcal{H}'$ equivalent to $\mathcal{H}$. Therefore, in the future, we plan to work on a mapping algebra that could include operators such as $\mathcal{S}$*-join*, $\mathcal{S}$*-meet* and $\mathcal{S}$*-Project*. Developing such operators allow us to restructure hierarchies given a set of mappings while preserving semantics.

REFERENCES

Bergamaschi, S., Castano, S., and Vincini, M. (1999). Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59.

Bouquet, P., Ehrig, M., Euzenat, J., Franconi, E., Hitzler, P., and et al. (2004). D2.2.1 specification of a common framework for characterizing alignment. Technical report.

Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J. D., and Widom, J. (1994). The TSIMMIS project: Integration of heterogeneous information sources. In *16th Meeting of the Information Processing Society of Japan*, pages 7–18, Tokyo, Japan.

Dey, A. and Abowd, G. (1999). The context toolkit: Aiding the development of contextaware applications. In *Dey, A.K. and G.D. Abowd. The Context Toolkit: Aiding the Development of ContextAware Applications. In Proceedings of Human Factors in Computing Systems: CHI 99. Pittsburgh, PA: ACM Press. pp. 434-441, May 15-20 1999.*

Fellbaum, C. (1998). *Wordnet: An Electronic Lexical Database*. ed. MIT Press.

Guarino, N. (May 1998). Some ontological principles for desigining upper level lexical ressources. In *Proc. of the First International Conference on Language Resources and Evaluation*, Granada, Spain.

Kashyap, V. and Sheth, A. P. (1996). Semantic and schematic similarities between database objects: A context-based approach. *VLDB Journal: Very Large Data Bases*, 5(4):276–304.

Madhavan, J., Bernstein, P. A., and Rahm, E. (2001). Generic schema matching with cupid. In *The VLDB Journal*, pages 49–58.

Magnini, B. and Cavaglia, G. (2000). Integrating subject field codes into wordnet. In *Proceedings of Language Resources and Evaluation (LREC 2000)*, pages 1413–1418.

Rahm, E. and Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350.

Sheth, A. P. and Kashyap, V. (1993). So far (schematically) yet so near (semantically). In *Proceedings of the IFIP WG 2.6 Database Semantics Conference on Interoperable Database Systems (DS-5)*, pages 283–312. North-Holland.

Sheth, A. P. and Larson, J. A. (1990). Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv.*, 22(3):183–236.

Stoœmer, H. and Stecher, R. (2005). An approach for context-based schema integration in virtual information environments. In *Doctoral Consortium in CONTEXT 05 - Fifth International and Interdisciplinary Conference on Modeling and Using Context, Paris - France*.

Torlone, R. and Atzeni, P. (2001). A unified framework for data translation over the web. In *WISE (1)*, pages 350–358.

Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., and Hübner, S. (2001). Ontology-based integration of information — a survey of existing approaches. In Stuckenschmidt, H., editor, *IJCAI–01 Workshop: Ontologies and Information Sharing*, pages 108–117.

Wiederhold, G. (1992). Mediators in the architecture of future information systems. *Computer*, 25(3):38–49.

Yaacoubi, N. and BenAhmed, M. (2003). Integrating smart communities in knowledge portals. In *IKE*, pages 523–528.