Distributed and Parallel Databases 7, 7–35 (1999)

Ontologies are not the Panacea in Data Integration: A Flexible Coordinator to Mediate Context Construction

ARIS M. OUKSEL aris@uic.edu
IQBAL AHMED
The University of Illinois at Chicago, Department of Information and Decision Sciences (M/C 294), Chicago, IL 60607

Recommended by: Athman Bouguettaya

Abstract. Shared ontologies describe concepts and relationships to resolve semantic conflicts amongst users accessing multiple autonomous and heterogeneous information sources. We contend that while ontologies are useful in semantic reconciliation, they do not guarantee correct classification of semantic conflicts, nor do they provide the capability to handle evolving semantics or a mechanism to support a dynamic reconciliation process. Their limitations are illustrated through a conceptual analysis of several prominent examples used in heterogeneous database systems and in natural language processing. We view semantic reconciliation as a nonmonotonic query-dependent process that requires flexible interpretation of query context, and as a mechanism to coordinate knowledge elicitation while constructing the query context. We propose a system that is based on these characteristics, namely the SCOPES (Semantic Coordinator Over Parallel Exploration Spaces) system. SCOPES takes advantage of ontologies to constrain exploration of a remote database during the incremental discovery and refinement of the context within which a query can be answered. It uses an Assumption-based Truth Maintenance System (ATMS) to manage the multiple plausible contexts which coexist while the semantic reconciliation process is unfolding, and the Dempster-Shafer (DS) theory of belief to model the likelihood of these plausible contexts.

Keywords: ontology, semantic reconciliation, heterogeneous database systems, heterogeneous information systems, semantic interoperability, data integration, semantic conflicts

1. Introduction

Organizations have witnessed an increasing demand for access to data obtained from multiple information systems to support operational control and decision making requirements. In this environment, the ability to effectively integrate data has a direct impact on the organizational performance. If the various information systems have been developed independently, they are likely to be semantically heterogeneous, in that, they differ in logical interpretations of data and domains captured by each system, in structural representations, in data models, and in naming and format specifications of data. A central problem of data integration is the design of mechanisms to resolve these semantic conflicts.

Several approaches [1, 2, 4, 6, 7, 19, 27], each with severe limitations, have been proposed in the past to deal with semantic conflicts. The most recent one, and arguably the most popular, is based on the use of a shared ontology, which is defined as a description of concepts and the relationships that can exist for an agent or a “community of agents” [6]. An ontology specifies a vocabulary to enable mutual understanding among its users as they adopt it to describe and to interpret shared information [5–7, 19]. For example, consider the following portion of the **Time** section from the Planning Ontology Initiative in the ARPA/Rome Laboratory Planning Initiative (ARPI) [26].

	Time
Duration	(Infinity, Epsilon, Zero, years, weeks, days, hours, minutes)
Duration-bounds	(min-duration, max-duration)
Time-point	(Date (Origin, Offset)
	(Calendar-date (year, month, day, hour, minute))
	pt-distance (pt<, pt<=, pt>, pt>, pt>=, pt=))
Time-interval	(Always, interval-begin, interval-end, min-duration, max-duration, interval-before)

The resolution of semantic conflicts using ontology is a three-step process [6]. First, a shared ontology is defined; second, objects from cooperating information systems are described using the ontology, and third, the similarity between different objects is determined via the ontology. For example, consider objects **periodicity**, and **time-factor** which exists in two financial databases DB1 and DB2, respectively. Assume that instances of ‘periodicity’ in DB1 represent the number of days for which a stock is associated with a particular risk factor (e.g., an instance ‘4’ of ‘periodicity’ refers to four days), while the instances of ‘time-factor’ in DB2 represent the time period in hours during which a stock is associated with a particular risk factor (e.g., an instance ‘4’ of ‘time-factor’ refers to a 4 hour period). On the basis of an *ontological commitment*, which is an agreement to use a vocabulary (i.e., ask queries and make assertions) in a way that is consistent with respect to the theory specified by an ontology [5], database administrators (DBAs) of these databases can map the objects ‘periodicity’ and ‘time-factor’ from their respective databases to the common concept **duration** in the ontology. A further mapping of ‘periodicity’ and ‘time-factor’ to *days* and *hours* respectively (both are also concepts present in the ontology) will ensure that an accurate semantic interpretation of these concepts is represented in the ontology. The ontology is then used to ascertain that object ‘periodicity’ and object ‘time-factor’ from DB1 and DB2 are semantically related.

Several promising semantic reconciliation approaches including the Concept Hierarchy Model (CHM) [27], the Summary Schema Model (SSM) [2], the Semantic Proximity Model (SemPro) [7], and the Context Interchange Network (COIN) [4, 19] are based on the ontology approach. SSM and CHM use the terms ‘summary schema’ and ‘concept hierarchy’, respectively, to describe those data structures used for reconciling semantic conflicts. According to the definition of ontology, both these data structures could be classified as ontologies. The concept hierarchy in CHM could in fact be categorized as a domain specific ontology.

The general assumption is that the ontology approach significantly reduces the complexity of determining similarity between different *objects* from Heterogeneous Information

Sources (HIS). This assumption is mainly justified on the basis of the criteria used to design ontologies [25]: (a) *Clarity*; the intent of the terms in an ontology is clear and there is minimal ambiguity amongst terms, (b) *Coherence*; the ontology is internally consistent, (c) *Extensibility*; the design of an ontology ensures that additional specialized terms can be added to the existing vocabulary without mandating an alteration of existing definitions, (d) *Minimal Encoding Bias*; the ontology is implemented to permit knowledge sharing across agents that may be implemented using different representation systems and styles, and finally, (e) *Ontological Commitment*; the ontology mandates the minimal ontological commitment required to facilitate the intended knowledge sharing exercise. It is our contention that ontology-based reconciliation suffers from the same kind of drawbacks as the global schema approaches [2, 6–8, 18], even if the above design criteria are actually satisfied. Specifically:

(a) It is neither practical nor theoretically possible to develop and maintain an ontology that strictly adheres to the design criteria specified above, particularly in an environment of evolving semantics [24].
(b) An ontology does not identify and classify semantic conflicts accurately.
(c) An ontology cannot handle query-directed reconciliation, which requires multiple interpretations of semantic conflicts.
(d) An ontology does not provide the coordination mechanism to discover metadata knowledge for semantic reconciliation, and to ensure consistency across all the mappings relevant to a query.

We illustrate these limitations through a conceptual analysis of two promising ontology-based semantic reconciliation techniques, namely COIN [4, 19] and SemPro [7]. Despite their drawbacks, ontologies are very useful in semantic reconciliation when their application is properly circumscribed. For example, they can serve to establish initial mappings, referred heretofore as **anchors**, between terms of a local and a remote information system. These initial mappings are necessary to constrain the propagation of search for the correct query-dependent semantic reconciliation between terms. In this paper, we assume that semantic reconciliation is generally context-dependent, and the set of interpretations is not commensurable, as is usually the case in most interesting applications. Therefore, interpretations of concepts, or across a multiplicity of concepts cannot be fixed a priori. We describe a semantic reconciliation system which circumvents the limitations of ontologies listed above, namely SCOPES (Semantic Coordinator Over Parallel Exploration Spaces) [18], and yet exploits their capabilities to initialize construction of the *context* used for reconciliation. **MIKROKOSMOS** [9, 10, 12], an ontology currently available online on the World Wide Web, is used for illustration.

In SCOPES, semantic reconciliation is a nonmonotonic reasoning process, where interschema mappings asserted at one point may be corroborated or retracted as further supporting or contradictory evidence is uncovered. Ontologies can be used to constrain exploration of a remote database during the incremental discovery and refinement process of the *context* within which a query to a remote database can be answered. The management of these multiple, often contradictory and yet plausible set of assertions is supported in SCOPES by using an Assumption Based Truth Maintenance System (ATMS) [3]. The

Dempster-Shafer (DS) theory of belief [20] is used in conjunction with the ATMS to model the likelihood of these plausible sets of assertions. An advantage of using ontologies, such as MIKROKOSMOS, is their provision of similarity measures that can be used to compute the measures of belief assigned to the multiple plausible contexts.

In Section 2 we describe a running example used throughout the paper. We motivate our work by proving through counter-examples the inadequacy of ontologies to deal with complex semantic reconciliation in general, and consequently justifying the design of a system like SCOPES in Section 3. In Sections 4 and 5, we give an overview of COIN and SemPro, examine why they fail to provide a solution to the problem of semantic heterogeneity, and speculate on how to increase their effectiveness in achieving semantic heterogeneity among HIS. In Section 6 we describe SCOPES, illustrate how SCOPES can overcome the drawbacks of ontologies and exploit their advantages. We conclude with a summary of results achieved and a brief list of further work.

2. Running example

The schemas described below will be used throughout the paper to illustrate our claims. Figure 1 is a partial schema of a database which maintains information on the Engineering Faculty of Chicago-based Universities, while figure 2 depicts a partial schema of an Employee database of Engineering-related firms. We will refer to these databases as DB1 and DB2, respectively.

- **DB1:** A database of Engineering Faculty members of Chicago-based Universities
  Data Model: Non-Normalized Relational Schema (partial), see figure 1.
- **DB2:** A database of employees of engineering-related firms
  Data Model: Non-Normalized Relational Schema (partial), see figure 2.

There exist several semantic correspondences between DB1 and DB2. First, class 'Faculty' in DB1 and class 'Employee' in DB2 intersect. Instances of attribute 'SS#' in DB1 correspond to instances of attribute 'ID' in DB2 where the employees are consultants from Chicago-based Universities. Attributes 'Dept' in DB1 and DB2 share some common domain values; as do 'Sal_Type' in DB1 and 'Comp_Type' in DB2; and 'Sal_Amt' in DB1 and 'Comp' in DB2. As we shall see later, these three pairs may be considered either as

Faculty (<u>SS#</u>, Name, Dept, Sal_Amt, Sal _Type, Affiliation, Sponsor, University....)

Faculty:	*Any tuple of the relation Faculty, identified by the key SS#*
SS#:	*An identifier, the social security number of a faculty member*
Name:	*An identifier, Name of a faculty member*
Dept:	*The academic or non-academic department to which a faculty member is affiliated*
Sal_Amt:	*The amount of annual Salary paid to a Faculty member*
Sal_Type:	*The type of salary such as Base Salary, Grant, and Honorarium*
Affiliation:	*The affiliation of a faculty member, such as teaching, non-teaching, research*
University:	*The University where a Faculty member is employed*

Figure 1.

Employee (ID, Name, Type, Employer, Dept, CompType, Comp, Affiliation....)

Employee:	*Any tuple of the relation Employee, identified by the key ID*
ID:	*An identifier, the social security number of an Employee*
Name:	*An identifier, Name of an employee*
Type:	*An attribute describing the job category of an Employee, such as Executive, Middle Manager, Consultant from another firm, etc*
Employer:	*Name of the employer firm such as AT&T, Motorola, General Motors, etc*
Dept:	*Name of the department where an Employee works*
CompType:	*The type of compensation given to an employee, such as Base Salary, Contract Amount*
Comp:	*The amount of annual compensation for an employee*
Affiliation:	*Name of the Consultant firm, such as a University Name, Andersen Consulting, ...*

Figure 2.

synonyms or as homonyms depending on the nature of the query. Attributes 'Affiliation' in DB1 and DB2 are homonyms, as are attribute 'University' in DB1 and attribute 'Employer' in DB2, because their domains do not overlap. The fact that the domains are not overlapping is simply circumstantial, and therefore it cannot be assumed a priori. The two attributes could as easily have been overlapping in other database instances.

Attribute 'University' in DB1 and 'Affiliation' in DB2 may be considered as synonyms for the subset of class 'Employee' where 'Employee.Type = Consultant' , and where the values in the domain of the attribute 'Affiliation' in DB2 correspond to the names of Chicago-based Universities. Semantic reconciliation approaches, such as COIN or SemPro, designed to identify and reconcile semantic incompatibilities, should be capable of discovering these distinctions. Yet, as we show in the next section, COIN and SemPro fail to capture the nuances necessary to properly classify the correspondences. In our view, no ontology is capable of modeling the complexity of possible semantic interpretations.

3. Problem motivation

Ontology-based reconciliation techniques require an ontological commitment, which implies a standardized use of concepts and a priori semantic knowledge before semantic incompatibilities can be resolved. We believe it is this standardization which limits the usability of the ontology approach. By statically fixing semantic interpretation as is done in a global schema approach, ontologies do not provide the flexibility necessary to handle query dependent integration between autonomously and independently designed information systems. For example, the terms 'University' and 'Employer' from DB1 and DB2, respectively, may both be mapped to a common concept **Employer** in an ontology. Thus, per the ontology, we may assert that these terms are synonyms. The assertion is true if a query: Q: "*List Names of Employers of Engineering Related Professionals*" is posed against the two databases. But if a query: Q': "*List Names of Academic Institutions*" is posed instead, the synonymy relation between the two terms no longer holds, and its assumption will produce erroneous results.

Terms 'affiliation' and 'affiliation' from DB1 and DB2, respectively, may be mapped to a term **Affiliation** in the ontology, thus asserting that these two terms are synonymous. This assertion is incorrect since the attributes corresponding to these two terms in DB1 and DB2 have different domain values, the synonymy relation asserted by the ontology is incorrect.

A cursory analysis of the schemas described in figures 1 and 2 reveals that term ‘affiliation’ in DB2 has a synonymy relationship with ‘University’ in DB1, instead. As it were, this latter correspondence will not emerge in an ontology-based semantic reconciliation process, because it will be pair (University, Employer) that will map to the same concept in a general ontology. One may speculate as to the possibility of constructing an ontology specifically designed to handle semantic reconciliation amongst DB1 and DB2. The pair (affiliation, university) may then be mapped to the same concept. This ontology may not show a synonymy relationship between the two terms ‘affiliation’. But, clearly, this obviates the generality and the flexibility of the semantic reconciliation autonomy of the two procedures, and imposes stringent restrictions on the two databases akin to a global schema approach. In addition to the impracticality of such an approach, the autonomy of databases is also violated.

One other way is to assume that the ontologies are designed autonomously. The semantic conflicts may then be resolved by composing the two ontologies. But composition of ontologies is also a complex semantic reconciliation problem. Furthermore, it has to be done dynamically each time a query is posed, otherwise it may suffer from the same problems as a global schema approach.

Another example is that of the generalization/specialization relationship between (Faculty, Employee) pair. Ontologies are useless in this case, since they can at best provide a similarity mapping between the terms without specifically identifying the exact nature of the semantic conflict, i.e., the generalization/specialization abstraction.

Reconciliation techniques such as COIN and SemPro attempt to resolve the semantic conflicts existing at the schema level by providing a standardized description of schema level objects at a higher level of abstraction, i.e., the *metaattribute* level. The metaattributes are additional concepts that explicitly specify the various semantic explications of the schema level objects [4, 7, 19]. The assumption underlying this approach is that semantic conflicts may occur only at the schema level and not at the metaattribute level, implying that:

1. There is a complete and universal agreement about the meaning and use of metaattributes across the network [24].
2. Data Base Administrators (DBAs) know a priori all possible contexts in which their database can be queried. As a result they specify at the outset all possible properties and their possible values for each attribute in their local schemas.

The first assumption suggests that even though different DBAs may assign different names to the schema level objects representing the same real-world objects, any particular property or characteristic of these objects is well understood in a standardized manner across any large network, will not change over time, and is universally mapped to the same concept(s) in the ontology by DBAs across the network. The approach overlooks the fact that it is the difference in perspectives of the same DBAs that is a major cause of the existence of semantic conflicts in the first place. The result is creating a structure similar to the global schema with the difference that semantics are now being ‘fixed’ at one higher level of abstraction, i.e., the metaattribute level instead of the schema level. Additionally, as shown above for the attributes *Affiliation* of DB1 and DB2, the values assigned to these metaattributes also represent real-life concepts, and unless these are ‘fixed’, also are susceptible to the same kind of semantic conflicts faced at the schema level.

The effect of the second assumption is again the creation of a structure akin to that in a global schema. For example, the DBA of DB2 may not explicitly associate concept *research* as a metaattribute the concept ‘Employee’, some instances of which also represent in DB2 the academics working for the company as consultants. Hence, a query posed against DB1 and DB2 inquiring about a list of researchers may not be answerable unless the DBA of DB2 knows a priori that such an information request may be of interest to someone in the network, and in anticipation of such a query associates concept *research* with concept *Employee* well in advance. Several researchers [7, 16, 18] have contended that this second assumption is stringent in most practical situations, since a priori inventory of all the possible ways a database may be queried is totally unrealistic.

The strength of a dynamic semantic reconciliation approach lies in its extensibility, scalability, and its capability of dealing with evolving semantics [4, 18]. It should be able to:

- identify and classify all semantic conflicts;
- support flexible query-dependent interpretation of semantic conflicts;
- support discovery and reconciliation of semantic conflicts in an environment of incomplete and uncertain semantic knowledge;
- support the coexistence of multiple plausible contexts during the reconciliation process; and
- support a query-directed coordination mechanism for the dynamic elicitation of semantic knowledge.

SCOPES possesses all the characteristics listed above. The same cannot be said of ontology-based approaches. Using two specific ontologies, namely COIN and SemPro, and a few counter-examples, we prove our claim. A detailed analysis of other ontology-based approaches is given in [15, 16].

4. Context interchange network

4.1. Brief description

The Context Interchange Network (COIN) [4, 19] is an approach which attempts to resolve the semantic heterogeneity problem by enumerating additional standardized terms, taken from a shared ontology, to describe the semantics of schema level terms. A context mediator is responsible for matching terms from different databases and resolving any remaining conflicts. The mediator in COIN is based on Wiederhold’s context mediator concept [23], and utilizes a theory of *semantic values* as a unit of data exchange among heterogeneous information systems (HIS). A **semantic value** is defined as a simple value combined with its related context, where a simple datum is an instance of a type; and the context of a simple value is defined as the associated metadata relating to its meaning, properties (such as source, quality, precision etc.), and organization. For example, consider the attribute ‘Sal_Amt’

Create table **Faculty**

(SS#	*char (11)*
Name	*char (40)*
Dept	*char (25) (Chairperson char (40), Size int, Budget float (Currency char (4), Scale factor int))*
Sal-Amt	*float (Currency char (9), Scale factor int, Type char (20))*
Affiliation	*char (20)*
Sponsor	*char (50) (Grant float (Currency char (4), Scale-factor int, term char (10)), Dept char (25))*
University	*char (60) (President char (40), Size int, Annual Budget float (Currency char (4), Scale factor int))*

Create table **Employee**

(ID	*char (9)(Type char (10))*
Employer	*char (15) (Head office char (40), Size int, Annual Budget float (Currency char (4), Scale factor int))*
Name	*char (38)*
Dept	*char (20)(Head char (38), Size int, Dept Code char (5), Location char (35))*
Comp	*float (Type char (6), Currency char (9), Scale factor int)*
Affiliation	*char (40) (title char (10), Head char (40), Size int)*

An example of the semantic value schemata for the above relations as defined by a predicate is given as:

Create scene for Faculty and Employee by predicates: Currency = 'dollars' and scale-factor = '1000'

Figure 3.

from DB1 that appears as a simple value ‘80’ in the database. The context of this simple value will be metadata such as its currency, scale factor, periodicity, etc. Its semantic value can be written as ‘80’ (currency = ‘dollars’, scale factor = ‘1000’, periodicity = ‘yearly’, type = ‘base salary’), where ‘currency’, ‘scale-factor’, ‘periodicity’, and ‘type’ are relevant concepts derived from the associated metadata. The terms ‘dollars’, ‘1000’, ‘yearly’, ‘base salary’ are the values assigned to metaattributes.

For the relational model the semantic values are modeled as a sub-tuple of the simple values. Each property of the semantic value corresponds to an attribute in the sub-tuple. These attributes are called *metaattributes*. The attributes present in the base tuple are called *base attributes*. For example, the attribute ‘Sal_Amt’ in figure 3 is a base attribute, whereas ‘currency’, ‘scale factor’, ‘periodicity’, and ‘type’ are metaattributes.

An example of **data environment** is illustrated in figure 3. These environments are created in accordance with a shared ontology to associate the relevant context with the simple values. The data environments consist of two components namely the *semantic-value schema*; which is a declaration of the properties associated with each semantic value, and the *semantic-value schemata*; which specifies values for these properties.

4.2. Limitations of the COIN approach

COIN has been called a dynamic integration approach in [4, 19]. Yet, as we show in the next subsections, a few counter-examples demonstrate that, despite reliance on a shared-ontology, COIN does not exhibit the characteristics of a truly dynamic approach. Let us consider the example in figure 4.

4.2.1. COIN is limited in its identification and classification of semantic conflicts. Consider the example in figure 4, where Q1 posed against DB1 is to be translated into a query Q2 against DB2. To construct Q2, COIN must find in DB2 the semantic entities corresponding to terms ‘SS#’, ‘Faculty’, ‘Sponsor’, and ‘University’ in Q1. Assume the ontology yields

Consider Q1 posed against DB1, and Q2 it's semantically equivalent translation posed against DB2.
Q1: 'List those 'UIC' faculty members who are consultants at Motorola'

Q1: Select SS# **From** Faculty **Where** Sponsor = 'Motorola' **and** University ='UIC'
Q1': Select ID **From** Employee **Where** Employer in ('Motorola', 'UIC')
Q2: Select ID **From** Employee **Where** Employee.type ='CONSULTANT' **and** Affiliation = 'UIC' **and** Employer ='Motorola'

Figure 4.

the following most likely synonymy correspondences: (SS#, ID), (Faculty, Employee), (Sponsor, Employer), (University, Employer) between the terms in Q1 and terms in DB2, respectively. Query Q1 can then be simply erroneously translated to Q1′. At this point, the context mediator does not possess sufficient intelligence to validate, refute, or refine the correspondences simultaneously across all terms asserted by the ontology, or to further investigate other correspondences, such as (University, Affiliation) pair, for an accurate translation of Q1 to Q2. The mediator has no mechanism to recover from initial plausible, yet incorrect, mappings. For all purposes, the process stops.

4.2.2. COIN lacks flexibility to interpret the query context. An advantage of COIN over traditional static integration approaches is its capability to support multiple contexts by using metaattributes. Still, the DBA's perspective and the ontology impose a priori the possible interpretations allowed. As a result, there are valid queries that cannot be handled. For example, consider the contradictory interpretations required by queries Q and Q′ as discussed in Section 3. COIN does not provide the flexibility to dynamically select the appropriate interpretation depending on the specific query, and thereby allow both Q and Q′ to be answered. The metaattributes of (University, Employer) pair are defined similarly in both databases. Hence, regardless of the context of a given query, such as Q′, the context mediator has no mechanism to refute the initial synonymy correspondence provided by the ontology.

4.2.3. COIN lacks any coordination mechanism. There are several possible semantic mappings between attributes 'SS#' in DB1 and 'ID' in DB2. The correct correspondence cannot be ascertained by confining the process of disambiguation to evidence directly tied to these two attributes only. Disambiguation in this case would require enlarging the investigation scope to the whole schematic environment to gain a better perspective. For example, the investigation could be pursued by examining the correspondence between objects 'Faculty' and 'Employee' to which the two attributes belong, or by looking at their instances. We refer to these investigation as "upward propagation" or "downward propagation", respectively. The ontology may be used to assign an initial mapping. Let us assume that objects 'Faculty' and 'Employee' are found to be homonyms or unrelated, the initial correspondence between 'SS#' and 'ID' must be revised as unrelated, and further knowledge should be elicited to discover other mappings to DB2. Similarly, refuting the synonymy relationship in pair ('University', 'Employer') may only be achieved by examining specific instances. For example, by determining that value 'UIC' of 'University' does not exist in the domain of 'Employer', the synonymy mapping is rejected. Note that the exact nature of the relationship between the domains is unnecessary for the query at hand. The disambiguation process is query-specific.

The two examples above clearly illustrate the necessity for a systematic coordination mechanism capable of managing the multiple plausible contexts that may coexist at any one point in time during the semantic reconciliation process in environment with incomplete and uncertain knowledge; of eliciting further evidence for their corroboration (refutation); and of updating the plausibility of these contexts as a result of knowledge discovery. COIN does not provide or support such coordination, and therefore is limited in its capability to deal with semantic interoperability amongst HIS.

5. The semantic proximity model

5.1. Brief description of SemPro

SemPro [7] attempts to capture the various semantic interpretations of a schema object by specifying additional descriptive terms taken from a shared ontology. A collection of these terms is considered as the context of an object. The representation of context in the SemPro model is of the general form:

$$\text{Context} = \langle (C_1, V_1)(C_2, V_2) \ldots (C_k, V_k) \rangle$$

Each C_i, where $1 \leq i \leq k$, is a contextual coordinate denoting an aspect of context. It may model some characteristic of the subject domain and can be obtained from a shared ontology; it may model an implicit assumption in the design of a database. And it may or may not be associated with an attribute A_j of an object O. Each V_i, where $1 \leq i \leq k$, can be a variable; it can be a set; it can be a variable associated with a context; and it can be a set associated with a context.

SemPro provides a mechanism for comparison and manipulation of different contexts, such as the context of a query and the definition context of an object. This mechanism is based on the specificity relationship between two contexts, which is defined as 'given two contexts C_1 and C_2, $C_1 \leq C_2$ iff C_1 is at least as specific as C_2. The specificity relationship induces a partial order such that a greatest lower bound (*glb*) exists for any two contexts thus leading to the organization of the context set as a meet semilattice. The glb (or the least common denominator in this case) of two contexts is defined as 'the most specific context which is more general than either of the two contexts.'

The representations in figures 5 and 6 are in accordance with the SemPro model of context. For example, in figure 5 the terms identifier, department, employer, sponsor,

Representation of Definition Context for Faculty and Employee Schemas

$\text{Faculty}_{\text{def}}$:< (identifier, {name, SS#}), (department,{DEPARTMENT}),
(reimbursement, {salary, honorarium, grant}), (affiliation, {research, teaching, non-teaching}
(employer, X o {UNIVERSITY}),(sponsor, {AT&T, Motorola, GM,....})>

$\text{Employee}_{\text{def}}$:< (identifier, {name, ID}), (Department, {DEPARTMENT}),(employer, {Motorola}),
(category, {Exec, MidMgr, CONSULTANT o <(affiliation, {firmname})>}),
(reimbursement, {Salary,Contract Amt.})>

Figure 5.

Representation of Q1's Definition Context

Q_{def}: < (identifier, SS#), (sponsor, Motorola), (employer, UIC), (Faculty, FACULTY) >

Figure 6.

reimbursement, category, affiliation are contextual coordinates. The terms in brackets are specified values for these contextual coordinates. '*X*' is a variable. UNIVERSITY and DEPARTMENT are objects. $X \circ$ {UNIVERSITY} denotes that X can take its values from the set of Universities. CONSULTANT is an object that is a subset of EMPLOYEE and has a contextual coordinate 'affiliation'.

5.2. Drawbacks of the SemPro approach

The intent of the above representation is to anticipate the likely mappings between different databases. The goal is to a priori circumscribe the set of possible contexts within which queries may be translated. The most specific context for a given query is then obtained by computing the 'glb' of the database context and the query context. The correlation of semantically related concepts in the resulting 'glb' is determined via an ontology. This approach may be efficient in limited application domains, since the set of contexts is finite. But, it is not adequate to deal with semantic conflicts in a dynamic environment with autonomous information sources. Basically the same conflicts as may have existed at the schema level have not been eliminated.

Consider the context definition for DB2 given by Employee_{def} in figure 5, and that for Q1 given by Q_{def} in figure 6. Consider once again mapping Q1, posed against DB1, to a semantically equivalent query against DB2. Figure 7(a) shows the resulting context, hereby referred to as C_{glb}. It was obtained using SemPro rules [7] for computing the glb of any two contexts. Figure 7(b) shows the context representation of the target query Q2 against DB2.

C_{glb} contains the knowledge that is required to translate Q1 to Q2. However, this translation can be obtained only if correct correspondences between DB1 and DB2 have been already determined. It is evident that due to the limitations already mentioned in the sections above, the *more general context* given by C_{glb} does not accurately identify all the

C_{glb} = glb (Employee_{def}, Q_{def}) :
<(identifier, {SS#}), (Department, {DEPARTMENT}),(employer, glb{Motorola, UIC}),(category, {Exec, MidMgr., CONSULTANT o <(affiliation, glb{firmname, research, teaching, non teaching})>}), (reimbursement, glb{Salary, Contract-Amt, grant }), (sponsor, Motorola), (Faculty, FACULTY) >

(a)

$Q2_{def}$: <(identifier, {ID),(employer, {Motorola}), (category, {CONSULTANT o <(affiliation, {UIC})>}),>

(b)

Figure 7.

correspondences nor does it provide the refinement process that can lead to a representation such as $Q2_{def}$.

In SemPro, it is assumed that all semantic interpretations of an object can be represented explicitly, thus ensuring the availability of all metadata knowledge needed for semantic reconciliation. While this is already questionable, it is precisely the discovery of these semantic interpretations that makes the problem of reconciliation very complex. For example, assuming C_{glb} contains within it the knowledge to translate Q1. There is no indication on how to discover this knowledge. SemPro does not provide a coordination mechanism that can be applied systematically to refine this context and to determine the precise knowledge, which is required to translate Q1 to Q2.

6. Semantic coordinator over parallel exploration spaces (SCOPES)

6.1. Lessons from COIN and SemPro

Both COIN and SemPro are useful to find anchors or points of reference in a remote database using ontology. These anchors are basically plausible mappings that are used to trigger further exploration of the remote database to ascertain the validity of these initial mappings or, if refuted, to discover new stronger mappings. Anchors serve as starting points for semantic reconciliation. The examples discussed above clearly illustrate the limitations of COIN and SemPro that impede the semantic reconciliation process in a dynamic environment. The mediation is highly dependent on a priori semantic knowledge, whose availability is unlikely in most practical situations. COIN and SemPro have an advantage over the global schema approach in that several plausible contexts are allowed to coexist, although one might argue that this is also possible in the global schema approach. The assumption is that these contexts can be defined a priori. But they do not display the flexibility to handle unanticipated queries and contexts in a dynamic environment as is likely to happen on the web or in intranets or extranets. Their dependency on the standardized use of concepts across a large network of databases is also unreasonable. The resolution of semantic conflicts mandates incrementally uncovering and piecing together the view of a remote schema, which is pertinent to answering a specific query [18].

6.2. Brief overview of the SCOPES approach

SCOPES [18] is an architecture, which facilitates incremental construction of the context within which meaningful exchange of information between HIS can take place. It supports semantic reconciliation under partial semantic knowledge, by coordinating query directed elicitation of information. Knowledge acquisition exploits available reconciliation techniques and other knowledge sources. Since the incremental system makes assertions under partial knowledge, the knowledge acquisition process may generate multiple plausible contexts, each of which needs to be corroborated through the acquisition of additional supporting evidence. The SCOPES approach incrementally uncovers and assembles together the specific view of another schema which is pertinent to answering a specific query. In our view, semantic reconciliation is a nonmonotonic reasoning process, where schematic mappings which are asserted at one point in reconciliation may be retracted after eliciting further evidence. In SCOPES the ATMS is used in conjunction with the Dempster-Schafer theory of belief for the representation and resolution of ambiguity.

6.3. Conceptual components used in SCOPES

6.3.1. Classification of semantic conflicts. Semantic conflicts between heterogeneous conceptual schemas arise due to differences in data models, differences in logical interpretation of data captured by each system, different structural representations, mismatched domains, and different naming and formatting schemes employed by the individual systems. We briefly describe below the classification of semantic conflicts [14] used in SCOPES.

Semantic conflicts are classified along three dimensions namely, *naming*, *abstraction*, and *levels of heterogeneity*. The Inter-Schema Correspondence Assertion (ISCA) which represents the semantic relationship between two elements of different databases has the general form:

Assert[naming, abstraction, heterogeneity]

where *naming* (*abstraction*) stands for a naming (abstraction) function between an element x in the local database and an element y in either the local or the remote database; heterogeneity indicates the structural schema description of x and y in their respective databases. This classification combines the dimensions of semantic conflicts with a structural description, the heterogeneity dimension, thereby facilitating the process of operational integration.

Along the dimension of naming, the relationships between two elements x and y can be categorized as *synonyms*, denoted $syn(x, y)$, which are terms having similar meaning; *homonyms*, denoted $hom(x, y)$, which are similar terms representing different concepts; and *unrelated* , denoted $unrel(x, y)$, which are not related along the dimension of naming; however, these could be related in some other way such as functional relationships. Along the dimension of abstraction, the relationships between two elements x and y can be categorized as *class* relationship, denoted $class(x, y)$; *generalization/specialization* relationship, $gen(x, y)$; *aggregation* relationships, denoted $agg(x, y)$; and relationships due to *computed* or *derived functions*, denoted $function\text{-}name(x, y)$. The levels of heterogeneity include the *object* level, the *attribute* level and the *instance* level of the database schema. Semantic conflicts due to naming and abstraction can occur at any of these levels within the same or two different databases. Thus this level provides us with a structural mapping between two corresponding elements from different databases. This dimension requires a pair of values, one for each element x and y, as represented in its corresponding schema. Each value is denoted either $att(x, O, DB)$, where x is the element considered in the assertion, O is the class of objects to which it is attributed, DB the database in which it appears; or $obj(x, DB)$, where x is a class in DB; $inst(x, O, DB)$, where x is an instance of class O in DB. (Note: when there is no ambiguity, we shall use the terms ‘object’ and ‘class’ interchangeably.)

The most important advantage of this classification is the partitioning of semantic conflicts into 12 disjoint classes based on the dimensions of naming and abstraction. Some of these classes are transient because they are valid only in the dynamic reconciliation environment, where they represent semantic conflicts which are classified in the presence of incomplete semantic knowledge, whereas the other classes are valid in both the static and the dynamic reconciliation environment. These disjoint classes are described in more detail in [14], where it is shown that the classification captures all semantic conflicts discussed in the literature on heterogeneous conceptual schemas.

6.3.2. Managing uncertainty and incompleteness. Consider again the example in figure 4. The translation of Q1 to Q2 requires mapping concepts 'SS#', 'Faculty', 'Sponsor', 'University', 'UIC', and 'Motorola' of DB1 to their equivalent ones in DB2. Let us now proceed to map the terms in query Q1 to their corresponding ones in DB2. Our first concern is to find **anchors** for the query terms in DB2. An ontology can play an important role in this process. Assuming that the concepts from both databases have been translated by their local DBAs to equivalent ones in the ontology, we can get an equivalence mapping via the ontology between the attribute 'SS#' of DB1 and 'ID' of DB2. Note that the mapping of local schema terms by the DBAs to concepts in the ontology are only initial, plausible mappings. SCOPES is capable of backtracking from an incorrect interpretation, as new evidence contradicting the initial mapping is discovered. An advantage of the SCOPES approach is that it provides a recovery mechanism in case any of the initially assumed mappings are incorrect with respect to the query's context.

Without any further knowledge about the interoperating databases, any of the 2^{12} subsets of assertions obtained from the set of 12 possible assertions is an equally valid context. To refine this set of assertions, further correspondences, such as the one between the objects 'Faculty' and 'Employee', to which the two attributes 'SS#' and 'ID' respectively belong, needs to be asserted. According to the context of the above query and in accordance with information available from the data dictionary in Section 2 , these pairs ought to be reconciled as synonyms where $Employee.Type = Consultant$ and $Employee.Affiliation = UIC$. Our intent is to illustrate how the classification of semantic conflicts described in the previous section, an ATMS, and a judicious use of an ontology, such as MIKROKOSMOS, are integrated in SCOPES to arrive at such conclusion. We use the DS theory of belief in conjunction with an ATMS to refine the context by corroborating and retracting our assertions as further evidence is collected, and to model the likelihood of these sets of assertions. We describe the salient features of the DS theory in the section below.

6.3.2.1. Models used for plausible semantic reconciliation. Given two elements from two different databases, there are 12 possible ISCAs based on the two dimensions of naming and abstraction. Let this set of assertions be Ω. There are 2^{12} possible subsets of Ω represented by the power set $P(\Omega)$. Recall that in SCOPES context is described as a set of plausible ISCAs. To represent an uncertain context, for example, the context representing the mapping between terms 'SS#' from DB1 and 'ID' from DB2 with respect to a query Q1, we use the DS theory to assign portions of belief committed to the subsets of Ω. The DS theory of belief is formulated in terms of a function:

$$m : P(\Omega) \rightarrow [0, 1] \quad \text{st:}\, m(\phi) = 0 \text{ and } \sum_{A \subseteq \Omega} m(A) = 1$$

The function m is referred to as a *mass function* or a *basic probability assignment*; every subset of the environment that has a mass value greater than 0 is called a focal element. Assume that while investigating a possible mapping between terms 'SS#' from DB1 and 'ID' from DB2, evidence E provides support for the following assertions:

A1: Assert[syn(SS#, ID), class(SS#, ID), (att(SS#, Faculty, DB1), att(ID, Employee, DB2))]
A2: Assert[syn(SS#, ID), gen(SS#, ID), (att(SS#, Faculty, DB1), att(ID, Employee, DB2))]
A3: Assert[syn(SS#, ID), agg(SS#, ID), (att(SS#, Faculty, DB1), att(ID, Employee, DB2))]

Let a mass function assign a value of 0.4 to the set {A1, A2} and 0.4 to the set {A3}. The left over mass value is assigned to the larger set $\Omega (m(\Omega) = 0.2)$ denoting that there may be additional conditions beyond evidence E which are true with some degrees of belief.

Definition (Evidence Set). Let Ω be the domain of values for a set of ISCAs. Evidence set (ES) is a collection of subsets of Ω associated with mass function assignments.

For example, in the above case $\text{ES}_1 = [\{\text{A1, A2}\}^{0.4}, \{\text{A3}\}^{0.4}, \Omega^{0.2}]$ is an evidence set.

Definition (Belief Function). A belief function, denoted by *Bel*, corresponding to a specific mass function m, assigns to every subset A of Ω the sum of beliefs committed exactly to every subset of A by m, i.e.,

$$Bel(A) = \sum_{X \subseteq A} m(X). \tag{1}$$

For example,

$$\begin{aligned} Bel\{\text{A1, A2, A3}\} &= m[\{\text{A1}\}] + m[\{\text{A2}\}] + m[\{\text{A3}\}] + m[\{\text{A1, A2}\}] + m[\{\text{A1, A3}\}] \\ &\quad + m[\{\text{A3, A2}\}] + m[\{\text{A1, A2, A3}\}] \\ &= 0 + 0 + 0.4 + 0.4 + 0 + 0 + 0 = 0.8. \end{aligned}$$

The above belief function is a measure of the minimum degree of support in favor of the set {A1, A2, A3}.

Definition (Plausibility Function). A plausibility function, denoted by *Pls*, corresponding to a specific mass function m, determines the maximum belief that can be possibly contributed to a subset of A, i.e.,

$$Pls = 1 - Bel(A^c) \tag{2}$$

where A^c is the complement of A in Ω, and is equivalent to $(\Omega - A)$. The plausibility function is defined to indicate the degree to which the evidence set fails to refute a subset A. For example,

$$Pls(\{\text{A1, A2, A3})\} = 1 - Bel(\{\text{A1, A2, A3}\}^c) = 1 - 0 = 1.$$

The plausibility function denotes the maximum degree to which the assertion set {A1, A2, A3} cannot be disproved and hence is plausible. We can observe that by definition, $Bel(A) \leq Pls(A)$. Their difference $Pls(A) - Bel(A)$ denotes the degree to which the evidence set is uncertain whether to support A or A^c.

Combining evidence sets: There may exist multiple evidence sets supporting different mass function assignments on a domain of values. Given two mass function m_1 and m_2 from two evidence sets ES_1 and ES_2, respectively, we can use the Dempster's rule of combination to combine them. The combined mass denoted by $m_1 \oplus m_2$ is defined as:

$$m_1 \oplus m_2(Z) = \sum_{Z=X\cap Y} m_1(X) \cdot m_2(Y) \tag{3}$$

For example, consider the availability of evidence set ES_2 which supports the following set of assertions

A1: Assert[syn(SS#, ID), class(SS#, ID), (att(SS#, Faculty, DB1), att(ID, Employee, DB2))]
A3: Assert[syn(SS#, ID), agg(SS#, ID), (att(SS#, Faculty, DB1), att(ID, Employee, DB2))]
A4: Assert[syn(SS#, ID), function(SS#, ID), (att(SS#, Faculty, DB1), att(ID, Employee, DB2))]

For clarity of expression let the mass function m corresponding to ES_1 above be denoted as m_1. The mass function corresponding to ES_2 is denoted by m_2 and assigns values 0.3 to the set {A3, A4}, and 0.6 to the set {A1}. The remainder of the mass value is assigned to the larger set $\Omega\,(m(\Omega) = 0.1)$. Hence, $ES_2 = [\{A3, A4\}^{0.3}, \{A1\}^{0.6}, \Omega^{0.1}]$. Table 1 shows how these two pieces of evidence can be combined in the DS theory to further refine the context. The values in the internal boxes are the result of the combination of evidence sets ES_1 and ES_2 using Dempster's rule of combining evidence. Since by definition $m_1 \oplus m_2(\phi)$ should be equal to 0, we need to normalize the internal values in the above table by using the following general formula:

$$m_1 \oplus m_2(Z) = \left[\sum_{Z=X\cap Y} m_1(X) \cdot m_2(Y)\right] \Big/ (1-k) \tag{4}$$

Table 1.

	$m_2[\{A1\}] = 0.6$	$m_2[\{A3, A4\}] = 0.3$	$m_2[(\Omega)] = 0.1$
$m_1[\{A1, A2\}] = 0.4$	{A1} 0.24	{ϕ} 0.12	{A1, A2} 0.04
$m_1[\{A3\}] = 0.4$	{ϕ} 0.24	{A3} 0.12	{A3} 0.04
$m_1[(\Omega)] = 0.2$	{A1} 0.12	{A3, A4} 0.06	(Ω) 0.02

where $k = \sum_{X \cap Y} = \phi\ m_1(X) \cdot m_2(Y)$. For example, in Table 1, $k = 0.12 + 0.24 = 0.36$ and $1 - k = 0.64$, the values in the boxes should be modified as follows:

$$m_1 \oplus m_2(\{\text{A1}\}) = (0.24 + 0.12)/0.64 = 0.56;$$
$$m_1 \oplus m_2(\{\text{A1, A2}\}) = 0.04/0.64 = 0.0625;$$
$$m_1 \oplus m_2(\{\text{A3}\}) = (0.12 + 0.04)/0.64 = 0.25;$$
$$m_1 \oplus m_2(\{\text{A4, A5}\}) = 0.06/0.64 = 0.093;$$
$$m_1 \oplus m_2(\Omega) = 0.02/0.64 = 0.031; \quad m_1 \oplus m_2(\phi) = 0.$$

6.3.2.2. Why the DS approach? The utility of probability theory for modeling reasoning with uncertainty is limited by the lack of sufficient data to accurately estimate the prior and conditional probabilities required in using Bayes' rule. The DS theory sidesteps the requirement for this data. It accepts an incomplete probabilistic model without prior or conditional probabilities. Given the incompleteness of the model, the DS theory does not answer arbitrary probabilistic questions. Rather than estimating the probability of a hypothesis, it uses belief intervals to estimate how close the evidence is to determining the truth of a hypothesis. When used to model sources of evidence that are not independent, it can yield misleading and counterintuitive results. The fact that the classification decomposes semantic conflicts into disjoint classes helps significantly in the process of avoiding errors. It is important to note that a nonmonotonic approach in accumulating assertions has provisions for retracting assertions and the DS approach can be used together with a nonmonotonic approach.

6.3.3. The necessity of the ATMS. The use of the DS approach requires an inference engine to deduce belief functions. We use an ATMS to provide a symbolic mechanism for identifying the set of assumptions needed to assemble the desired proofs, so when we assign probabilities of these assumptions, the system can be used as a symbolic engine for computing degrees of belief sought by the DS theory. The second important use of the truth maintenance system is to handle the effect of retracting assumptions when they are invalidated by the evidence and to keep track of the multiple plausible sets of assertions which can coexist in the absence of complete knowledge. Truth maintenance systems arose as a way of providing the ability to do dependency backtracking when assumptions or assertions are retracted because they are contradicted by the current knowledge, and so to support nonmonotonic reasoning. Nonmonotonic reasoning is an approach in which axioms and/or rules of inference are extended to make it possible to reason with incomplete information.

6.4. *The SCOPES solution illustrated*

Instead of presenting the solution formally, which will only obscure the method, we opt to illustrate how SCOPES works through an example, but give sufficient detail to explicate the unfolding algorithm. In SCOPES, a context is operationally defined as a set of consistent ISCAs between schema elements from two given databases, say DB1 and DB2. The ISCAs must be consistent with respect to available, and generally incomplete, evidence. As the evidence itself may be uncertain, so is the consistency of the context. An ATMS maintains all plausible contexts as the evidence is being gathered, and rejects only those contexts that are found to be inconsistent because of certain contradictory evidence. In this way the reconciliation process takes advantage of both supportive and contradictory evidence. Let us again consider our problem of mapping query Q1 to Q2, and specifically schema elements ‘SS#’ from DB1 and ‘ID’ from DB2.

Without any knowledge about the interoperating databases, any of the 2^{12} subsets of assertions obtained from the set of 12 possible assertions is an equally valid context. This set of contexts forms a complete lattice under the containment relationship. Semantic reconciliation in SCOPES is the process of incrementally gathering semantic or structural evidence to refine our ‘belief’ or ‘nonbelief’ in any of 2^{12} contexts. A specific piece of evidence supports exactly one context. It is the greatest lower bound (glb) of all contexts that this evidence satisfies. Evidence is gathered to incrementally piece together a view of the remote schema which in turn enables us to narrow down the search space from 2^{12} to a manageable and tractable number and further refine the context for reconciliation. A word of caution here to the performance-minded reader: operationally, only those plausible contexts with respect to evidence are maintained. Thus the manageability is directly proportional to the amount of knowledge gathered about a remote site.

6.4.1. The initial mappings. Terms in Q1 are mapped to terms in DB2 using knowledge sources such as reconciliation techniques, thesauri, concept hierarchies or ontologies. The success of the semantic reconciliation process is highly dependent on the availability of these latter sources of knowledge. The availability of an ontology-based reconciliation technique such as COIN or SemPro is advantageous since the ontology component of these approaches can be utilized in SCOPES to heuristically generate the initial correspondences between the two databases. However, the mappings obtained as a result of this exercise are not sufficient to guarantee a meaningful exchange of information among HIS. For example, in Section 3 we illustrated how the interpretations of queries Q and Q′ may be ‘fixed’ if we rely completely on an ontology for semantic reconciliation. Below we illustrate how SCOPES can exploit the advantages of ontologies in general while overcoming their limitations.

We asked two different DBAs to map the local schema objects from their respective databases, DB1 and DB2, to corresponding concepts in MIKROKOSMOS. Table 2 shows these mappings. Tables 3–6 show the correspondences established via MIKROKOSMOS between Ontology Concepts from DB1 and DB2. The respective similarity strength (score) given by MIKROKOSMOS for each mapping is also listed.

6.4.2. Algorithm

Step 1: Initialization

Semantic reconciliation in SCOPES is query-directed. The first step is to determine the semantic relevance of a remote database with respect to a specific information request. A

Table 2.

Concepts from DB1	Corresponding MIKROKOSMOS concepts for DB1	Concepts from DB2	Corresponding MIKROKOSMOS concepts for DB2
SS#	Identify	ID	Identify
Faculty	University-faculty	Employee	Support-staff
University	University	Affiliation	Affiliate
Sponsor	Sponsor	Employer	Organization
Motorola	For-profit-organization	Consultant	Advising-entity
UIC	Academic-organization	Dept	Organization-division
Affiliation	Affiliate	Compensation	Salary-attribute
. . .	. . .	. . .	. . .

Table 3.

DB1 ontology concept	DB2 ontology concepts	Similarity score
Identify	Identify	1.0
Identify	Organization	0.46
Identify	Affiliate	0.35
Identify	Advising-entity	0.27
Identify	Support-staff	0.24
Identify	Salary-attribute	0.118

Table 4.

DB1 ontology concept	DB2 ontology concepts	Similarity score
University-faculty	Organization	0.68
University-faculty	Affiliate	0.45
University-faculty	Advising-entity	0.40
University-faculty	Support-staff	0.40
University-faculty	Identify	0.29
University-faculty	Salary-attribute	0.13

Table 5.

DB1 ontology concept	DB2 ontology concepts	Similarity score
Sponsor	Organization	0.719
Sponsor	Organization-division	0.547
Sponsor	Affiliate	0.545
Sponsor	Advising-entity	0.422
Sponsor	Support-staff	0.30
Sponsor	Identify	0.27

methodology such as one proposed in [13] can be integrated within SCOPES to facilitate this process. Once the semantic relevance with respect to a query is established, for example, DB2 is selected as relevant, the semantic reconciliation process can then proceed.

Step 2: Mapping query terms

To find correspondences for the query terms in the remote database, the coordination algorithm in SCOPES uses the MIKROKOSMOS ontology as a heuristic. For each concept in the local database, select the correspondence with the next highest similarity not already considered or rejected. For example, the synonymy mapping between terms ‘SS#’and ‘ID’ (i.e., between concepts ‘identify’ and ‘identify’ from Table 3) is considered first. In the absence of any further evidence, any ISCA between attributes “SS#” from DB1 and “ID” from DB2 is possible. It is reasonable to assume the synonymy relationship between these two terms to be valid. In our ATMS, this mapping will be considered as a separate context; let it be denoted as C1. SCOPES assumes C1 to be the most plausible context with respect to available evidence.

Step 3: Schema propagation to expand exploration of context

SCOPES pursues the validation of C1 either by *upward propagation* in the schema structure to gather evidence on the structural relationship between the entities to which the two attributes respectively belong; or by *downward propagation* in the schema structure to gather comparative evidence on the relationship between their respective domain values. Several other propagation techniques [8] are utilized to expand the exploration of the context. SCOPES can then utilize available knowledge sources such as ontology-based reconciliation techniques, to either elicit a mapping between the terms representing the entities, or to compare the domains of the two attributes. In our example, we continue using MIKROSMOS. From Table 4, this exploration via the ontology results in the following evidence: (Note again: terms ‘object’ and ‘class’ interchangeably).

E1: syn(term(Faculty, DB1), term(Employee, DB2)) with probability 0.4

Generally, evidence such as E1 may be obtained using a number of knowledge sources including ontologies, lexicons, reconciliation techniques, general or domain specific knowledge repositories, metadata specifications, general rules derived from conceptual structures etc. [18]. The strength of this evidence is determined by its source. For example, in the case above a probability of 0.4 is assigned to E1 based on the similarity score provided by MIKROKOSMOS in Table 4 for concepts (University-Faculty and Support-Staff).

Step 4: ISCAs inference

E1 is used below to further narrow the search space. In SCOPES the reconciliation techniques and knowledge sources are coordinated using the following interface template:

$$\text{r: IF C(p) THEN consequent. } [p.q] \tag{5}$$

where the antecedent of the rule C is defined recursively in BNF (Backus-Naur Form) as follows:

$$C ::= E \mid \text{Assertion} \mid \text{Assumption} \mid E \text{ and } C \mid \text{Assertion and } C \mid \text{Assumption and } C$$

Additionally, C may include quantifiers over the domains of the variables. As a result, ‘C’ can be an extremely complex typed predicated logic expression constructed from either a directly elicited piece of evidence ‘E’, available knowledge, a reasonable assumption, or a combination thereof. In the above template “consequent” is a disjunction of assertions about two objects O1 and O2, p represents the degree of belief in all the assertions in ‘C’, and q is the degree of belief in rule r if $p = 1$.

There may be several rules that derive the consequent of rule r. The DS theory provides a feature called *parallel reduction* to derive the degree of belief in the consequent. However, all the matching rules need not be activated at the same time. Actually, selection of rules to activate is determined by various optimization strategies. A discussion of these strategies is beyond the scope of this paper. Let us assume the availability of the rules listed below. These rules are general, and thus independent of any specific database or application domain. (The following notation is used: Obj(O, t) denotes that Object ‘O’ is represented by term ‘t’; term(t, O) denotes that ‘t’ is the term corresponding to object ‘O’; att(t, O) denotes that term ‘t’ represents an attribute of object ‘O’; dom(O) denotes the domain of object ‘O’; key(t, O) denotes that ‘t’ is the term corresponding to the key of object ‘O’.

r1: **IF** syn(t, t′) ∧ Obj(O, t) ∧ Obj(O′, t′) **THEN** syn(O, O′) ∧ (gen(O, O′) ∨ agg(O, O′) ∨ class(O, O′))
r2: **IF** syn(O, O′) ∧ att(t, O) ∧ att(t′, O′) ∧ key(t, O) ∧ key(t′, O′) **THEN** syn(t, t′)
r3: **IF** syn(O, O′) ∧ att(t, O) ∧ att(t′, O′) ∧ not(key(t, O)) ∧ not(key(t′, O′)) **THEN** syn(t, t′) ∨ hom (t, t′) ∨ nrelated(t, t′)
r4: **IF** gen(O, O′) ∧ term(t, O) ∧ term(t′, O) **THEN** gen(t, t′)
r5: **IF** dom(O) ≠ dom(O′) **THEN** hom(O, O′)
r6: **IF** $\forall v_1 \in$ dom(O), $\exists v_2 \in$ dom(O′) ∧ syn(v_1, v_2) **THEN** syn(O, O′)
r7: **IF** syn (O, O′) ∧ class(O, O′) ∧ att(t, O) ∧ att(t′, O′) ∧ key(t, O) ∧ key(t′, O′) **THEN** syn(t, t′) ∧ class(t, t′)
r8: **IF** syn(O, O′) ∧ gen(O, O′) ∧ att(t, O) ∧ att(t′, O′) ∧ key(t, O) ∧ key(t′, O′) **THEN** syn(t, t′) ∧ gen(t, t′)
r9: **IF** syn(O, O′) ∧ agg(O, O′) ∧ att(t, O) ∧ att(t′, O′) ∧ key(t, O) ∧ key(t′, O′) **THEN** syn(t, t′) ∧ agg(t, t′)

The above rules clearly comply with the interface template. For simplicity, the degree of belief in each of the above rules is assumed to be $q = 1$. This is obviously an overestimation. While this may reduce the performance of the system in general because of potentially increasing backtracking during reconciliation, it avoids the perennial argument in expert systems of whether any probability assignment conforms to reality even if it is provided by experts. A rule r whose consequent is a disjunction of literals, as is rule r1, may be decomposed into as many rules as there are literals in the consequent. Each of the resulting rules will have the same premise as r and exactly one literal as consequent. The degree of belief of the original rule is then uniformly distributed over all the resulting rules.

E1 matches rule r1. The consequent leads to A1, A2 and A3.

A1: Assert[syn(Faculty, Employee), class(Faculty, Employee), obj(Faculty, DB1), obj(Employee, DB2)]
A2: Assert[syn(Faculty, Employee), gen(Faculty, Employee), obj(Faculty, DB1), obj(Employee, DB2)]
A3: Assert[syn(Faculty, Employee), agg(Faculty, Employee), obj(Faculty, DB1), obj(Employee, DB2)]

$$\text{A} = \{\text{A1, A2, A3}\}, m(\{\text{A1, A2, A3}\}) = p * q = 0.4$$
$$\text{where } p = 0.4 \text{ and } q = 1.0, m(\Omega) = 0.6$$

Applying Eqs. (1) and (2): $Bel(A) = 0.4$ and $Pls(A) = 1$.

With a degree of belief of 0.4 we can say that A1, A2, and A3 are part of the relevant context. SCOPES can now use this belief to refine the context for (‘SS#’, ‘ID’) pair.

Step 5: Derivation

Evidence E2 can be deduced from the set {A1, A2, and A3}.

E2: syn(obj(Faculty, DB1), obj(Employee, DB2)) [0.4]

This is yet another example of how SCOPES gathers evidence during the reconciliation process, i.e., the consequent of rules becomes part of the body of collected evidence.

Step 6: Repeat Steps 4 and 5 until no derivation

Evidence E2 matches part of the premise in rule r2. The rest of the premise simply requires schema information in DB1 and DB2, which can readily be obtained with degree of belief of 1.0. The consequent of rule r2 enables us to generate the following evidence:

$E2'$: syn(att(SS#, Faculty, DB1), att(ID, Employee, DB2)) [0.4]

This degree of belief for $E2'$ is obtained by using Eq. (5). But information obtained from the ontology (see Table 3) indicates that the two attributes are synonyms with similarity score equal to 1.0. In other words, the synonymy relationship is certain. Obviously, this fact overrides any other on this matter generated by inference. This case illustrates again another situation where the availability of an ontology helps narrow the search process. The belief degree of $E2'$ is updated to 1.0, and thus:

A4: Assert[syn(SS#, ID), class(SS#, ID), (att(SS#, Faculty, DB1), att(ID,Employee, DB2))]
A5: Assert[syn(SS#, ID), gen(SS#, ID), (att(SS#, Faculty, DB1), att(ID, Employee, DB2))]

$$A = \{\text{A4, A5}\}, m\{\text{A4, A5}\} = 1.0 \text{ and } m(\Omega) = 0.0$$

Applying Eqs. (1) and (2) $Bel(A) = 1.0$ and $Pls(A) = 1.0$.

This is an example of **context merging**, which allows us to extrapolate consistent interpretations when combining plausible sets of assertions, derived from the mapping between two different yet schematically related pairs of objects.

Step 7: Context merging

If no additional evidence is available, the two contexts provided by the assertion sets {A1, A2, A3} and {A4, A5} cannot be refined further. However, since these two sets represent assertions about objects which are not entirely independent, i.e., 'SS#' is a key attribute of 'Faculty' and 'ID' is a key attribute of 'Employee', a pair-wise combination of members of these sets can be further investigated. We refer to this process as **context merging**. Context merging is a method to refine a query context by combining contexts dealing with schema related terms and eliminating those combinations that are contradictory. For example, consider the context {A1, A2, A3} for pair (Faculty, Employer) and {A4, A5} for pair {SS#, ID}. The resulting, potentially consistent, contexts are {A1, A4}, {A1, A5}, {A2, A4}, {A2, A5}, {A3, A4}, and {A3, A5}). According to rules r7, r8, and r9 in the rule base, only sets {A1, A4} and {A2, A5} are consistent since all other sets contain contradictory assertions. Figure 8 is a pictorial representation of the belief network resulting from the refinement of context discussed above.

Step 8: Repeat Steps 2 to 7 until all query terms are processed

SCOPES repeats the above reconciliation process for the other terms in Q1: from Table 6 the ontology yields a plausible synonymy mapping between terms 'University' in DB1 ('University' in Table 6) and 'Employer' in DB2 ('Organization' in Table 6). Again, this mapping is viewed in our ATMS as one separate context. Let this latter context be denoted C1′. Any ISCA between terms 'University' in DB1 and 'Employer' in DB2 is valid in the absence of further knowledge. Once again 2^{12} subsets may be considered. Heuristically, because of the high similiarity measure, SCOPES assumes C1′ as the most likely context

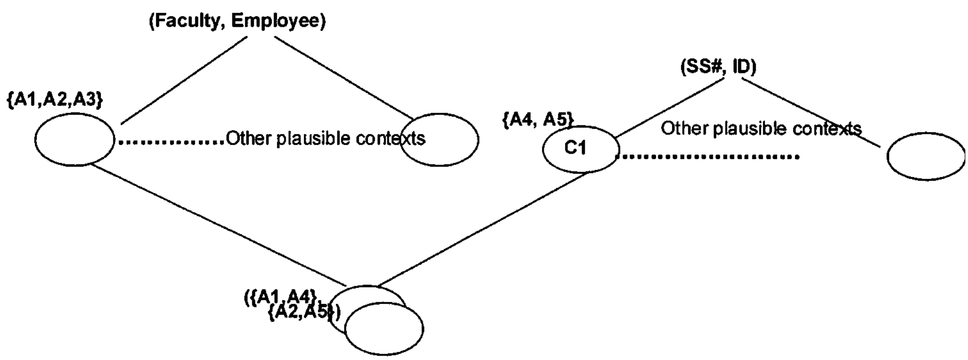

Figure 8.

Table 6.

DB1 ontology concept	DB2 ontology concepts	Similarity score
University	Organization	1.0
University	Organization-division	0.67
University	Advising-entity	0.52
University	Affiliate	0.4
University	Support-staff	0.31
University	Identify	0.25

and triggers further exploration through upward and downward propagation. Since the objects to which the two terms respectively belong are 'Faculty' and 'Employee', evidence E2 combined with the fact that 'University' and 'Employer' are not keys of their respective objects triggers rule r3. The consequent of rule 3 does not provide any discrimination along the naming dimension, and therefore, the search space remains potentially of size 2^{12} plausible contexts. In this situation, ontology knowledge is extremely useful. Knowledge from the ontology can be used as a heuristic to obtain evidence E3, which in turn triggers r1 and results in set {A6, A7, and A8}.

E3: syn(att(University, Faculty, DB1), att(Employer, Employee, DB2)) [1.0]

A6: Assert[syn(University, Employer), class(University, Employer),
(att(University, Faculty, DB1), att(Employer, Employee, DB2))]
A7: Assert[syn(University, Employer), gen(University, Employer),
(att(University, Faculty, DB1), att(Employer, Employee, DB2))]
A8: Assert[syn(University, Employer), agg(University, Employer),
(att(University, Faculty, DB1), att(Employer, Employee, DB2))]

$$A = \{\text{A6, A7, A8}\},\ m\{\text{A6, A7, A8}\} = p * q = 0.7$$
$$\text{where } p = 1.0 \text{ and } q = 0.7,\ m(\Omega) = 0.3$$

Applying Eqs. (1) and (2): $Bel(A) = 0.7$ and $Pls(A) = 1$.

If no additional evidence is available, the contexts provided by the assertion sets {A1, A2, A3} and {A6, A7, A8} cannot be refined further. However, since 'University' is an attribute of 'Faculty' and 'Employer' is an attribute of 'Employee', context merging of members of these sets is investigated. Context merging results in the following pairs of equally likely context assertions: {A1, A6}, {A1, A7}, {A1, A8}, {A2, A6}, {A2, A7}, {A2, A8}, {A3, A6}, {A3, A7} and {A3, A8}). It can be confirmed using our rule base that all of the above sets are valid interpretations since they do not contain contradictory assertions. Note, however, there are several situations where a computed degree of belief for each of the above contexts may be different. If it is the case, the degrees of belief are used to rank the contexts and to prioritize context selection for further processing, i.e., depth first search. The computation of the degrees of belief for the contexts resulting from context merging is beyond the scope of this paper.

Step 7b: Generate further evidence for same terms and compute new degree of belief using DS

SCOPES has the capability to take advantage of the query structure using available local schema information. For example, 'UIC' exists as a domain value of attribute 'University' in DB1. The latter knowledge maybe automatically elicited from a tool such as the multi-database language [8]. The coordination algorithm in SCOPES exploits this knowledge to investigate the domain of 'Employer' in DB2 to find a match for 'UIC', through downward propagation. Assume the following evidence is uncovered.

E4: dom(University) $\neq$ dom(Employer) [1.0]

The Evidence E4 matches r5 and this rule is fired. The consequent r5 enables us to conclude that the set A of plausible assertions is A9.

A9: Assert[hom(University, Employer), class(University, Employer),
att(University, Faculty, DB1), att(Employer, Employee, DB2))]

$$A = \{\text{A9}\}, m\{\text{A9}\} = p * q = 1.0 \quad \text{where } p = 1 \text{ and } q = 1.0, m(\Omega) = 0.0$$

Applying Eqs. (1) and (2): $Bel(A) = 0.8$ and $Pls(A) = 1$.

This is again a situation where Dempster's rule of combination can be applied to determine the combined belief resulting from evidence sets E3 and E4. This is illustrated below in Table 7.

The values in the internal boxes of Table 7 are the result of the combination of evidence sets E3 and E4. Since by definition $m_1 \oplus m_2(\phi)$ should be equal to 0, we need to normalize the internal values using Eq. (4). The values in parentheses represent the normalized values.

The assertion sets {A9} and {A6, A7, A8} are two plausible alternatives. Assertion A9 contradicts assertions A6, A7 and A8. If the belief in A9 is 1.0 then the ATMS will mark assertions A6, A7, and A8 as retracted. However, the current belief in A9 is 0.54 does not allow for the retraction of the set {A6, A7, A8} whose degree of belief is now 0.318. SCOPES chooses the one with the higher degree of belief to continue the reconciliation process, i.e., depth-first search. The set {A6, A7, A8} is temporarily retracted, until and only when reconciliation is shown impossible with the assumption of {A9}.

If no additional evidence is available, the contexts provided by the assertion sets {A1, A2, A3} and {A9} cannot be refined further. Context merging results in the following pairs:

Table 7.

	$m_2[\{\text{A9}\}] = 1.0$	$m_2[(\Omega)] = 0.0$
$m_1[\{\text{A6, A7, A8}\}] = 0.4(0.318)$	$\{\phi\}$ 0.4	{A6, A7, A8} 0.0
$m_1[(\Omega)] = 0.6(0.682)$	{A9} 0.6 (0.54)	(Ω) 0.00

{A1, A9}, {A2, A9}, {A3, A9}. It can be confirmed using our rule base that all of the above sets are valid interpretations since they do not contain contradictory assertions.

Since A9 asserts a homonymy relationship between objects 'University' of DB1 and 'Employer' of DB2, it becomes necessary to find a new mapping for query term 'University' of DB1. SCOPES utilizes the next alternative mapping provided by MIKROKOSMOS in Table 6. The next two rounds of reconciliation consider successively concepts 'Department' and 'Consultant' from DB2. Each of these cases are results again in a homonymy relationship being asserted in a similar fashion as was done for concept "Employer" above. The concept 'Affiliation' of DB2 is considered next. The validation process generates the following evidence;

E5: syn(att(University, Faculty, DB1), att(affiliation, Employer, DB2)) [0.4]

A10: Assert[syn(University, Affiliation), class(University, Affiliation), (att(University, Faculty, DB1), att(Employer, Employee, DB2))]
A11: Assert[syn(University, Affiliation), gen(University, Affiliation), (att(University, Faculty, DB1), att(Employer, Employee, DB2))]
A12: Assert[syn(University, Affiliation), agg(University, Affiliation), (att(University, Faculty, DB1), att(Employer, Employee, DB2))]

We update our belief as follows:

$$A = \{\text{A10, A11, A12}\}, m\{\text{A10, A11, A12}\} = p * q = 0.4$$
$$\text{where } p = 0.4 \text{ and } q = 1.0, m(\Omega) = 0.6$$

Applying Eqs. (1) and (2): $Bel(A) = 0.4$ and $Pls(A) = 1$.

As described earlier, the coordination algorithm in SCOPES exploits the availability of any schema knowledge through upward and downward propagation; i.e., University = 'UIC', to further refine context. SCOPES will trigger reconciliation by searching domain values of 'Affiliation' to find a match for 'UIC'. Assume the following evidence is uncovered:

E6: 'UIC' $\in$ dom(University, DB1) $\wedge$ 'UIC' $\in$ dom(Affiliation, DB2) [0.6]

The Evidence E6 matches r6's premise resulting in it being fired. The consequent of r6 enables us to conclude with a confidence level of $m = 0.6$ (since $p = 0.6$ from the evidence and $q = 1.0$) the set A of plausible assertions contains A10. $Bel(A) = 0.6$, $Pls(A) = 1.0$. This is again an example of a situation where Dempster's rule of combination can be applied to determine the combined belief resulting from evidence sets E5 and E6, as follows:

The values in the internal boxes of Table 8 are the result of the combination of evidence sets E5 and E6. Since by definition $m_1 \oplus m_2(\phi) = 0$, we do not need to normalize the above values. $m_1 \oplus m_2$ ({A10}) = 0.7; $m_1 \oplus m_2$ ({A10, A11, A12}) = 0.16; $m_1 \oplus m_2$ $(\Omega) = 0.24$; $m_1 \oplus m_2$ $(\phi) = 0$.

The assertion sets {A10} and {A10, A11, A12} present us with two plausible alternatives. Assertion A10 is also a member of {A10, A11, A12}. SCOPES selects the set with the higher

Table 8.

	$m_2[\{A10\}] = 0.6$	$m_2[(\Omega)] = 0.4$
$m_1[\{A10, A11, A12\}] = 0.4$	{A10} 0.24	{A10, A11, A12} 0.16
$m_1[(\Omega)] = 0.6$	{A10} 0.36	(Ω) 0.24

degree of belief, which in this case is {A10}. Once this selection is made, the set {A10, A11, A12} is no longer under consideration, however, since A10 is under consideration, only assertions A11 and A12 are marked as temporarily retracted in the ATMS.

If no additional evidence is available, the contexts provided by the assertion sets {A1, A2, A3} and {A10} cannot be refined further. Context merging results in the following pairs: {A1, A10}, {A2, A10}, {A3, A10}. It can be confirmed using our rule base that all of the above sets are valid interpretations since they do not contain contradictory assertions. The belief network in figure 9 below illustrates this context refinement process.

A process similar to one described above can establish a synonymy correspondence between 'Sponsor' in DB1 and 'Employer' in DB2. Assume that context merging in this case results in the following sets: {A1, A13}, {A2, A13}, {A3, A13} where A13 asserts that:

A13: Assert[syn(Sponsor, Employer), class(Sponsor, Employer),
(att(Sponsor, Faculty, DB1), att(Employer, Employee, DB2))]

Context merging involving all the query terms, and taking **in consideration all** the **intermediate context** merging steps such as in figure 8, results in the following two sets: {A1, A4, A10, A13} and {A2, A5, A10, A13} which are sufficient to map Q1 to Q2.

7. Conclusions

Shared ontologies are not a definitive solution to semantic reconciliation problems as several researchers have claimed. Through a conceptual analysis of two promising ontology-driven

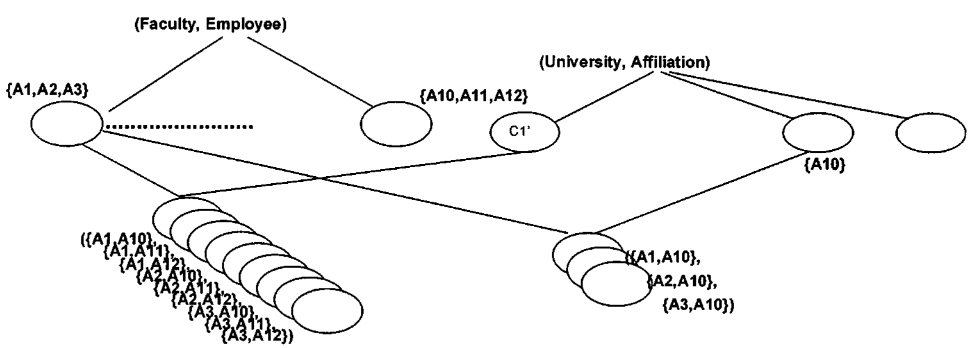

Figure 9.

semantic reconciliation techniques, namely COIN and SemPro, we have demonstrated their deficiencies as the sole interface between semantically heterogeneous information systems. Ontologies are useful, but they do not alone resolve semantic conflicts. We presented arguments for the set of properties needed to build truly dynamic semantic reconciliation systems. Essentially, the robustness of semantic reconciliation system rest on their capabilities to be query-directed, handle multiple interpretations in an environment of incomplete and uncertain semantic knowledge, and adapt to evolving semantics. We have described, however briefly, a system that possesses these properties, namely the SCOPES architecture, and have shown the appropriate use of a general ontology (MIKROKOSMOS). The examples demonstrated how the advantages of ontologies are exploited and their drawbacks overcome within a general coordination mechanism.

There are several issues not address here in part due to the scope of the paper. A nonexhaustive list includes: (i) performance issues, which basically deal with the pruning of the search space; other types of semantic conflicts, such as temporal, spatial and causal. An important aspect of the system is its learning capabilities. Each time a query is processed a lot of knowledge is learned by the system which may not be of direct benefit to the query at hand, but will be useful in subsequent queries assuming no significant changes occurred to the databases. We are also extending the classification to capture spatiotemporal conflicts.

References

1. C. Batini, M. Lenzerini, and S. Navathe, “A comparative analysis of methodologies for database schema integration,” ACM Computing Surveys, vol. 18, no. 4, 1986.
2. M.W. Bright, A.R. Hurson, and S. Pakzad, “Automated resolution of semantic heterogeneity in multidatabases,” ACM Transactions on Database Systems, vol. 19, no. 2, pp. 212–253, June 1994.
3. J. DeKleer, “An assumption-based truth maintenance system,” Artificial Intelligence, vol. 28, pp. 197–224, 1986.
4. C. Goh, M. Siegel, and S. Madnick, “Context interchange: Overcoming the challenges of large scale interoperable database systems in a dynamic environment,” in Proceedings of the 3rd International Conference on Knowledge Management, Maryland, 1994, pp. 337–346.
5. T.R. Gruber, “A translation approach to portable ontology specifications,” Knowledge Acquisition, vol. 5, no. 2, pp. 199–220, 1993. http://www-ksl.stanford.edu/kst/what-is-an-ontology.html
6. J. Kahng and D. McLeod, “Dynamic classificational ontologies: Mediators for sharing in cooperative federated database,” in Proc. 1st IFCIS International Conference on Cooperative Information Systems (CoopIS ’96), Brussels, Belgium, June 1996, pp. 26–35.
7. V. Kashyap and A. Sheth, “Schematic and semantic similarities between database objects: A context-based approach,” VLDB Journal, vol. 5, no. 4, October 1996.
8. W. Litwin, I. Mark, and N. Roussoupoulos, “Interoperability of multiple autonomous databases,” ACM Computing Surveys, vol. 22, no. 3, 1990.
9. K. Mahesh and S. Nirenburg, “A situated ontology for practical NLP,” in Proc. Workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, August 1995.
10. K. Mahesh and S. Nirenburg, “Meaning representation for knowledge sharing in practical machine translation,” in Proc. FLAIRS-96 Track on Information Interchange, Florida AI Research Symposium, May 1996.
11. E. Mena, V. Kashyap, A. Sheth, and A. Illarramendi, “OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies,” in Proceedings of the 1st IFCIS International Conf. on Cooperative Information Systems (CoopIS ’96), Brussels, Belgium, June 1996, pp. 14–25.
12. MIKROKOSMOS: http://crl.nmsu.edu/Research/Projects/mikro/ontology/onto-intro-page.html

13. S. Milliner, A. Bouguettaya, and M. Papazoglou, “A scalable architecture for autonomous heterogeneous database interactions,” in Proc. the 21st International Conference on Very Large Databases, VLDB ’95 Proceedings, Zurich, Switzerland, September 1995.
14. C. Naiman and A. Ouksel, “A classification of semantic conflicts,” Journal of Organizational Computing, vol. 5, no. 2, pp. 167–193, 1995.
15. A. Ouksel and I. Ahmed, “Plausible inference of context in heterogeneous information systems,” in Proc. 5th International Workshop on Information Systems and Technology, Amsterdam, December 1995, pp. 120–129.
16. A. Ouksel and I. Ahmed, “Coordinating knowledge elicitation to support context construction in cooperative information systems,” in Proc. 1st IFCIS International Conference on Cooperative Information Systems (CoopIS ’96), Brussels, Belgium, June 1996, pp. 4–13.
17. A. Ouksel and I. Ahmed, “Using SCOPES for dynamic construction of context in heterogeneous information systems,” in Proc. AIS 1996 Americas Conference on Information Systems, Phoenix, Arizona, August 1996, pp. 623–625.
18. A. Ouksel and C. Naiman, “Coordinating context building in heterogeneous information systems,” Journal of Intelligent Information Systems, vol. 3, pp. 151–183, 1994.
19. E. Sciore, M. Siegel, and A. Rosenthal, “Using semantic values to facilitate interoperability among heterogeneous information systems,” ACM Transactions on Database Systems, pp. 254–290, June 1994.
20. G. Shafer, (1976). A Mathematical Theory of Evidence, Princeton University Press, Princeton N.J.
21. A. Sheth and J. Larson, “Federated database systems,” ACM Computing Surveys, vol. 22, no. 3, 1990.
22. M. Siegel and S. Madnick, “A metadata approach to resolving semantic conflicts,” in Proc. 7th International Conference on Very Large Databases, September 1991, pp. 133–145.
23. G. Wiederhold, “Mediators in the architecture of future information systems,” Computer, vol. 25, no. 3, March 1992.
24. L. Wittgenstein, Philosophical Investigations, Macmillan Publishing: New York, NY, 1953.
25. WWW, Refer to: Design Criteria for Ontologies: http://www.dfki.uni-kl.de/~vega/CoMem/Ontology/onto-design-gruber/ subsection3_1_2.html 28 Feb 95
26. WWW, Refer to: Plan Ontology Construction Group (POCG): KRSL 2.0.2 Ontology Hierarchy: http://isx.com/pub/ARPI/ARPI-pub/krsl/ontology-hierarchy.txt
27. C. Yu, B. Jia, W. Sun, and S. Dao, “Determining relationships among names in heterogeneous databases,” SIGMOD RECORD, vol. 20, no. 4, pp. 79–80, December 1991.