Ontologies: an Approach to Resolve Semantic Heterogeneity in Databases

Farshad Hakimpour
Department of Geography
University of Zurich
farshad@geo.unizh.ch

Andreas Geppert
Department of Information Technology
University of Zurich
geppert@ifi.unizh.ch

1.0 Introduction

As the number of data providers and amount of data is increasing the integration and interoperability techniques are attracting attentions. Database integration is used to combine the data and interoperability is used for interaction between information systems and databases from different sources. A common issue in both approaches is *consistency.* A data set is considered consistent if all the user defined consistency constraints are satisfied by all the data items in the data set [Grefen, 1992] - a data set can be result of combining data from several systems or based on interaction between several systems in users' view.

Many consistency constraints are derived from semantics of the data items in the application domain - while some are not. For instance, referential key constraints are not defined based on the semantics of the objects. On the other hand a constraint such as is based on semantics of the objects building and railroad in the application domain.

We distinguish two main problem here: database consistency (based on the constraints) and semantics heterogeneity. Consistency constraints are derived from semantics of data and are used to validate the data. An example of database consistency constraint in a spatial database is: "railroads do not cross buildings". If all the railroads and buildings represented in a data set obey the constraint one can consider the data set consistent according to this constraint. Nevertheless, the original data producer could have considered railroads passing through tunnels under ground as "railroad"; while, the user of the data classifies such railroad under another class "underground". In this case, there is a semantic heterogeneity between the data producer's community (interpretation of the producer) and the user community (interpretation of the user), however, the data set can be qualified as a consistent data set according to the constraint. Semantic heterogeneity concerns users, not only, when deploying data from different providers, but also, when using user interfaces provided by different systems.

One aspect of MIGI[1] project is detection and resolution of semantic heterogeneities. It is important to make it clear what we mean by semantics. John Sowa defines semantics in [Sowa, 2000] as following:

> *"Semantics determines how the constants and the variables are associated with things in application domain".*

---

[1] The project is funded by Swiss National Science Foundation (SNSF). http://www.ifi.unizh.ch/dbtg/Projects/MIGI, Project Number: 2100-053995

By *semantics* we refer to the meaning of schema elements (e.g., classes, attributes and methods) and it is often used in contrast with *syntax*. By syntax we refer to the definition of the structure of schema elements. We consider syntax to be context independent. On the other hand, semantics is the people's interpretation according to their understanding of the world and therefore, context dependent. Different interpretation of data causes semantic heterogeneity. Relying on common sense is a critical source of semantic heterogeneity and explicit definition of how the data should be interpreted is a solution to this problem.

Schemas are the definitions of logical structures (or patterns) that convey the data and are the result of database design phase. Schemas are expressed in a language (known as Data Definition Language or DDL). Part of the semantics is based on the interpretation of DDL syntax - i.e., keywords, operators and their orders. That is, when encountering such keywords or operators a computer program takes a standard action or a human would have a standard interpretation. Another part of semantics is related to the names (or terms) one uses for identifiers in the DDL. Items in schema definitions such as: attributes, classes, methods, data types, relations are declared by specified names (or terms) and possibly some descriptions as metadata. Such verbal descriptions used to be the way to specify semantics of identifiers in schemas. The later part of semantics (terminological semantics) is the focus of this paper, while the former part is still subject of research (e.g., heterogeneity of OODB schemas and RDB schemas) it will not be discussed here.

Formalization of the terms used in the schema definitions is a way to clarify the terminological semantics. Explicit and formal definition of semantics of the terms guided many researchers to apply *formal ontologies* as a potential solution of semantic heterogeneity. In the domain of philosophy ontology explains the nature and essential properties and relations of all beings (Webster's Unabridged Dictionary) and is based on the truth and the nature of the beings independent of one's perspective of the world. As the primary property of all beings is their existence, ontology refers to philosophical investigation of existence or being. It can concern questions such as "What exists" and "What general sort of things are there."

In the domain of artificial intelligence formal ontology is defined as "explicit specification of conceptualization" by Gruber in [Gruber, 1993]. In this domain ontology has been used for sharing and reusing knowledge between agents with emphasis on the formalizing the specification of concepts and relations used by them. Formal ontologies also attracted attentions in integration of information systems and databases [Guarino, 1998b] - we adopt the definition by Nicola guarino [Guarino, 1998a] in this report.

Formal ontology is considered more than schema definitions in databases. Schemas are more concerned with the syntactic aspects of data representation and many researchers have been focusing on the syntactical aspects of schema integration. While schemas are used for organizing data in databases, formal ontologies are concerned merely with the understanding of the members of communities. It is important to note that schema definitions are based on the ontology definition (and vise versa). These definitions may also help one to estimate the ontology. However, schema definitions should commit to the domain ontology which is considered part of metadata [Staudt et al., 1999].

Applying formal ontologies to resolve semantic heterogeneity does not conclude defining a unique robust definition of terms and compel a community to have exactly such interpretation. By using formal ontologies communities are supposed to communicate based on their defined ontologies and complication in building ontologies is the cost of resolving ambiguity and semantics heterogeneity.

The structure of this report is as following. Section 2 aims to give a introductory definition of ontology applicable to the integration problems in database and information systems. Section 3 introduces some of the previous related work and experiences done on the Internet and their features. Section 4 discusses two major logic based representation formalism and discuses their abilities concerning representation of semantics and automated reasoning. Section 5 discusses the role of formal ontologies in an integration task.

2.0 Formal Ontology

Formal ontology [Guarino, 1998a] consists of *logical axioms* that convey the *meaning* of terms for a *particular community*. Logical axioms are the means to introduce concepts, relations and their taxonomic hierarchies, also express constraints. An ontology exists only under a consensus by members of a community [Bishr et al., 1999, Kottman, 1999] - e.g., users of one information system or people in one discipline.

The logical axioms mentioned above define explicit specification of *conceptualization* [Gruber, 1993]. Conceptualization is defined by a *domain* (D), a set of *states of the world* (W), and a set of *intensional* (or *conceptual*) relations ($\mathfrak{R}$) in [Guarino, 1998a]. The set of intensional relations introduced by Nicola Guarino [Guarino, 1998a] is a key issue in his definition. The following example can illustrate the above definition.

Consider states of the world illustrated in Figure 1. By the following conceptualization:

$D = \{A, B, C\}$ (Domain of our world)

$\mathfrak{R} = \{\rho_{Box}, \rho_x\}$ (Intensional or conceptual relations)

Set of intensional relations ($\mathfrak{R}$) map every state of the world to the following extensional relations:

$\mathbf{R} = \{R_{Box}, R_x\}$ (Extensional relations)

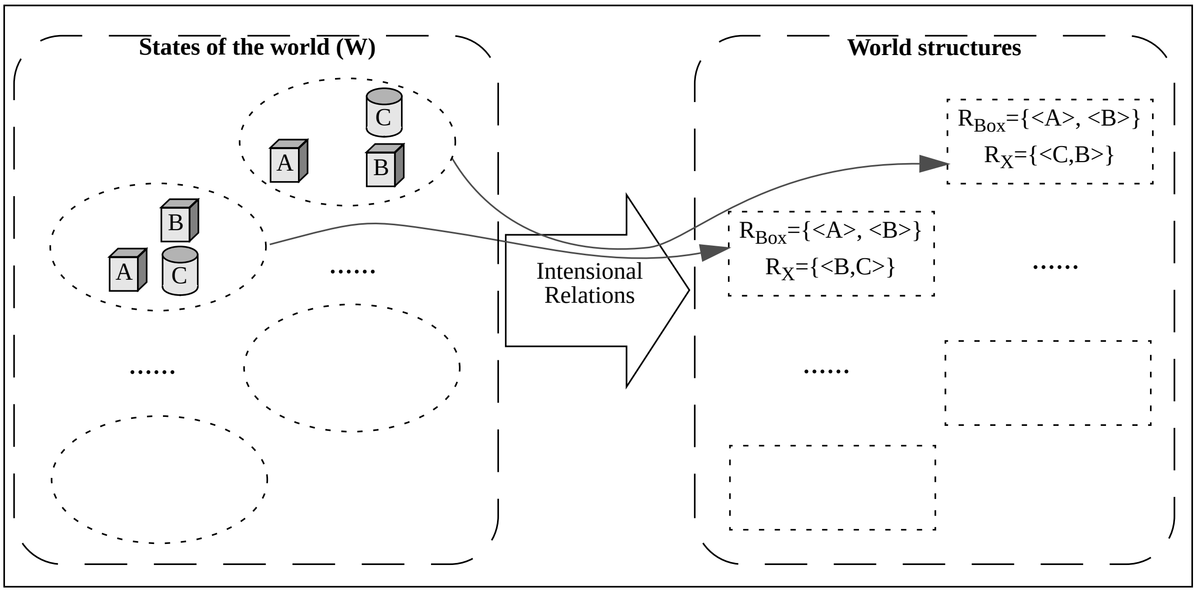

FIGURE 1. caption comes here

By some luck, one may guess what relation ρ_x is. (It is more like trying to answer a question "what is a book?" by showing some instances of books - the same way we learned it during our childhood.) By explicitly stating that such relation intends the meaning of "above", the relations will be more than only a Cartesian product of sets (e.g., DxD). By such definition of conceptualization, we are able to overcome the lack of the extensional relation to represent the semantics of the relation R_X. By such explicit definition we define a mapping from each state of the world to one particular subset of all possible tuples in the relation R_X [Guarino, 1998a]. A set of extensional relations (the subset of all possible tuples for every intensional relation) is a representation of the state of the world, and is called world structure (Figure 1).

Note that although individual A (in Figure 1) is closer to B and C in one state in compare to the other one, our extensional relation has not been influenced by such change. That is due to the fact that the intensional relation "close" is not defined in this conceptualization.

Consider the following axioms as a definition of the intensional relation "above":

above(x,y):

Hight(Center_of_Mass(x)) > Hight(Center_of_Mass(y)) and

Latitude(Center_of_Mass(x)) > Minimum_Latitude(y) and

Latitude(Center_of_Mass(x)) < Maximum_Latitude(y) and

Longitude(Center_of_Mass(x)) > Minimum_Longitude(y) and

Longitude(Center_of_Mass(x)) < Maximum_Longitude(y) and

....

In fact, all the new terms introduced in the definition have to be defined, as well - we rely on the common sense. After defining the intensional relation, one can assign the terms "over", "above" or "ueber" (or any other term) to this definition. Such mapping from used vocabulary of a community to intensional relations ($\Re$) or members of the domain (D) is considered *ontological commitment* in [Guarino, 1998a].

2.1 What Is not Formal Ontology

In contrast to a thesaurus, which applies a limited number of known relations among terms, an ontology describes the relations among the concepts referred by terms. Ontology plays a role in the conceptual level by defining concepts (Figure 2).

The definitions of concepts in an ontology will not exactly coincide with the concepts in the conceptualization. One can only try to roughly express the intensional relations. This inexactness is due to several reasons.

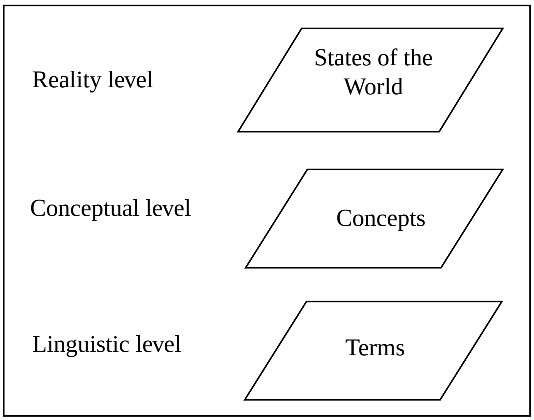

FIGURE 2. Caption

A concept as it is known in our thought can not be fully expressed by axioms. As an example a complete definition of "book" is not as easy as understanding the concept book. One may define "book" by the following axioms:

A book is made of paper.

It is printed.

It has rectangle shape.

It is bound in one edge.

These axioms are plausible. Yet, a book can be printed on plastic sheets and considered a book in our conceptualizations. A book can be written in handwriting, or have a hexagonal shape, as well. Even if one separates all the pages of a book we still consider it as a book. Many concepts have even more complication - e.g., time or space. What makes something *to be* truly a book has been investigated by philosophers and referred to by ontology.

Complication and ambiguity in the intensional relations can cause many difficulties in their formalized definitions. If not impossible, yet it is very difficult to be fully independent from the background knowledge or common sense. As shown in the above example, when defining a term, one needs to use other terms. This causes every concept definition depending on other concepts. Considering that one can not define all the terms used in the axioms, we have to rely on our common understanding of some basic terms - e.g, consider defining the IS_A relation (specialization relation). That is, relying on common sense is practically inevitable.

Consensus over definitions in an ontology among members of a community is required [Kottman, 1999]. This is an important difference between ontologies and conceptual models (or conceptual schemas) [Elmasri and Navathe, 2000]. While conceptual models are application dependent ontologies are only based on people’s understanding. Ontologies can be applied for the mapping between conceptual models.

The more detailed definitions in an ontology for a community makes it more difficult to reach a consensus within the community. That holds for building an ontology for several communities. For instance, by adding the following condition to our definition of “above” we try to avoid having overlap between the two individuals in the domain:

Minimum_Hight(x) > Maximum_Hight(y)

This axiom specializes the definition of “above” while all members may not agreed with adding such axiom - it is not an appropriate constraint to many contexts. Therefore, an ontology for a large number of communities cannot be highly specialized (or complete). Adding Constraints can specialize an ontology for a smaller community within a larger community or among several communities committing to a more general community. That is, a community can adopt a general ontology and specialize it by adding more terms. A definition may be added to an ontology only under the consensus of all members. That is no member can alter or override the definitions in an ontology according to his/her preferences - otherwise, there will be occurrence of hyponym - yet, one can add a new intensional definitions to specialize an ontology for a subcommunity [Visser et al., 1998].

Weaknesses in formalizations used to represent formal ontologies is another fact to consider in applying ontologies. Representation formalisms also impose their weaknesses over the axiomatization. As you will see in the next section every formalization has its own potential to express the intensional relations. It depends on the application ontology is used for.

In spite of the weakness in the above-mentioned axioms to express the intensional relations (such as “book” or “above”) they may be considered part of *an* ontology definition of the intensional relations “book” and “above” in *a* context - we also refer to such definition as *intensional definition*. Intensional definition can be used in application domains such as artificial intelligence and database integration. The main advantage of ontologies is that they help applications to be independent from the implicit background knowledge of the community or at least reduce the dependency on background knowledge of the community. One has to explicitly say what his or her intension is by referring to a defined term. This reduces the chance of semantic heterogeneity in communications among the communities - or their respective systems. Afterwards, the main concern would be finding out the discrepancies in extension of the concepts and avoid misinterpretation.

2.2 Philosophical Ontology and Formal Ontologies

The nature of the states of the world and the intensional (or conceptual) relations are not the concern of this report. There is no *thing* in the states of the world, that is, *thing*s exist after the conceptualization and we can formalize them based on the conceptualization. States of the world are subject to intensional relations. What people refer to by “real world” is the world structure (which is the result of mapping by intensional relations from states of the world). Study of the states of the world and their existence and nature is subject of philosophy. Also, investigation on the nature of the intensional relations or how different people conceptualize states of the world in the same or different ways and their validity is subject to philosophy and possibly psychology. All we are concerned with in this report is finding relations among the intensional relations and express them in axioms in term of other intensional relations. Our definitions in a formal ontology is expressing every intensional relation ρ_i by its relation to other intensional relations.

3.0 Related Works

As mentioned before formal ontologies attracted attentions in domain of data integration. We discuss the related works from three points of view which are related to this report. First perspective is related to building formal ontologies. This concerns questions such as “how do we distinguish a concept from a relation or an individual”, “how to build a taxonomy tree”, and “how good an ontology agrees conceptualization of a community”. The second view to research in this field is representation of ontologies and reasoning with and/or based on ontologies. Problems such as “how to formulate or represent an ontological definition” or “what feature are important in representing ontologies for reasoning” are addressed here. The third perspective concerns semantic integration of data or schema in databases. This topic addresses questions such as “how formal ontology can help to solve heterogeneity problems”, “what kind of heterogeneity problem can it solve” or “how does ontology interact with existing system architectures”. The related works are briefly discussed in the following from the three points of view mentioned above.

3.1 Building Formal Ontologies

Work of Guarino and Welty is one of the major contributions to the first area. They define notions such as rigidity, identity, unity, individuality, and dependence [Guarino and Welty, 2000a, Guarino and Welty, 2000b], such notions play important role in building ontologies. By applying those notions one can evaluate an ontology in term of its explication (i.e., how an ontology reveals implicit assumptions) and its accordance with the conceptualization of the community.

A good example of application of such work is to distinguish between designing a conceptual model and building an ontology. Using such notions introduced by Guarino and Welty, one can clarify source of heterogeneities caused by conceptual modeling in domain of special information systems. We used to classify spatial objects by primitive types: point (0D), line (1D) and area (2D). A spatial object inherits its representation depend on its dimension, a building is either a point or an aerial object or a street is a linear or aerial object. Such objects can change depend on the scale or type of application, while staying the same individual. Therefore, the dimension in which an object is represented is not a rigid property. Considering a building as a specialization of aerial object is not justified! In fact, building should be related to its representational dimension not inheriting from it [Guarino, 1999].

The same approach can be seen in COIN project [Goh et al., 1999] where “money amount” is considered a subtype of “semantic number” while number is only a primitive type for representing the value - or “currency type” is a subtype of “semantic string”. However, according to our definition of ontology it is based on conceptualization of people in a community. Therefore, “money amount” is an amount or a quantity. Treating “semantic number” as a supertype is the result of influence of application development. While, “money amount” or “currency type” are related to a value of type number or string, respectively, only for representation purpose.

Visser et al. in [Visser et al., 1999] illustrate a three phased approach they used to extract and build local ontologies from an application domain in KRAFT project. KRAFT is a project for integration of heterogeneous information using ontologies to resolve semantics problems. They propose an approach to build ontologies based on analysis of the technical texts available in the domain.

$(KA)^2$ [Benjamins and Fensel, 1998] is an initiative for building ontologies in knowledge acquisition community. The ontologies build by this initiative is used by OntoBroker [Benjamins and Fensel, 1998] for annotating Web pages and a better search facilities on Web.

Another well known project is Ontolingua a project on the on the Internet with the purpose of building ontologies. The Ontolingua web site supports users with software tools to create ontologies on the Internet and keeps a library of ontologies.

An important issue in this perspective is how to organize ontologies which has direct consequences on managing them. Hierarchical structure has been used as a proper way of organizing and managing ontologies in some of the related work [Heflin and Hendler, 2000, Visser and Cui, 1998, Visser et al., 1998] ([Farquhar et al., 1997] suggest more relations between ontologies). An ontology for a large number of communities cannot be complete or highly specialized. The more detailed definitions in an ontology for a community, the more difficult it is to reach a consensus within the community - or between communities. A community can adopt a *higher level* ontology (a more general but agreed upon ontology by several communities) and specialize it by adding more definitions to it. As a result, a specialized ontology cannot remove any constraint or term of a higher level ontology [Visser et al., 1998].

If a concept defined in a domain ontology has no relation to a higher level ontological definition, finding the similarity relation with another ontology cannot be done by a reasoning system.

3.2 Representing and Reasoning with Ontologies

Projects such as SHOE [Heflin and Hendler, 2000] and Ontobroker [Benjamins and Fensel, 1998] are using ontologies to improve the searching abilities on the World Wide Web. Both systems are using their own extension to HTML tags and logical reasoning based on ontological definitions.

In SHOE ontologies are created as a taxonomy hierarchy and a user queries the system by traversing the hierarchy. A web page on the Internet can specify an ontology and by adding tags defined in that specified ontology refers to the definitions in the ontology. SHOE Ontologies have the following elements in their definitions:

- Categories (concepts)
- Relationships (relations)
- Constants (they are particular instances needed generally for concept definitions)
- Inferences (constraints or rules)
- Definitions (they are verbal documents)

In SHOE, a Webcrawler looks for the Web pages with SHOE extension to HTML tags and keeps the extracted information in a knowledge base. This knowledge base also is used in turn to reply to a search query. A user should query the system by means of traversing the taxonomy hierarchy.

OntoBroker uses ontologies produced by $(KA)^2$ initiative [Benjamins and Fensel, 1998] - unlike SHOE, in which, several ontologies can exist and a web page may refer and commit to one. It keep a taxonomy hierarchy by means of IS-A relations (Just like SHOE) and represents attributes in their definitions. It is using a formalism similar to Frame Logic for reasoning. As a result the means to define relations between concept in Ontobroker is rule definition. Here are the main features in definitions for terms in On2Broker:

- is-a hierarchy of inheritance for terms (similar to hierarchy of concept definitions)
- attribute definition (similar to role definition or a relation with only type constraint)
- rules (not only considered as constraints but also as means of establishing relations)

OBSERVER [Mena et al., 1998] is another project using ontologies to allow queries against heterogeneous sources. It uses ontologies to replaces terms in user queries with suitable terms in target ontologies. OBSERVER uses Description Logic as both ontology definition language and query language and Classic performs the integration task.

Semantic Web is a frame work in W3 Consortium considering semantic issues on the Internet. RDF (model and schema) is an standard by the consortium. RDF has the necessary features to represent ontologies, however no reasoning system is available to work with this standard yet.

3.3 Ontology based Integration

There are comprehensive research work on the integration of databases specially concerned with the schema integration. Work of Kim et al. is a study for classification of schema heterogeneities [Kim et al., 1993, Kim and Seo, 1991]. Also, they present solutions for several types of schema heterogeneity in RDBs and OODBs. Another comprehensive work in this area is presented in [Garcia-Solaco et al., 1996]. Both works address problems of schema heterogeneity but do not distinguish between the schematic and semantic issues, while, we define the difference between the two and focus on the semantic problems here.

[Bergamaschi et al., 1998] introduced an approach to integrate schemas based on a thesaurus. They extract these relations from schema definitions of component databases - not directly from an ontology. They extract the same three relations used by OBSERVER. The approach is a semiautomatic relation extraction and needs supervision of an expert. Based on the extracted relations they introduce an algorithm to integrate schema definitions and produce a global integrated schema.

A module related to this perspective in OBSERVER [Mena et al., 1998] is the Inter-Ontology Relationships Manager. It keeps the relation between ontological definitions of terms in different ontologies. By means of such interontology relations, OBSERVER replaces terms in user queries with suitable terms in target ontologies. The inter-ontology relations are synonym, hypernym and hyponym (the same three relations used in [Bergamaschi et al., 1998]) which are also the relations used to build every ontology in OBSERVER. [Mena et al., 1998] refers to the translation of terms in user ontology based on inter-ontology relations as ontology integration.

KRAFT uses shared ontology [Jones, 1998] as a basis for mapping between ontology definitions and communication between agents. In [Visser et al., 1999], shared ontology is "*chosen to make shared ontology as expressive as the 'union' of the ontologies*". However, the definition of union of ontologies and its similatities or differences with shared ontology is not stated. KRAFT detects a set of ontology mismatches (as described in [Visser et al., 1998]) and establishes mapping between the shared ontology and local ontologies.

COIN project [Goh et al., 1999] is based on Domain Model. One can compare role of domain models with ontologies. However, a Domain Model is built in COIN more like a conceptual model and used as an ontology. COIN presents a good example of an architecture for semantic interoperability. The components of the architecture suit an ontology based approach. Articulation of data based on ontology and relating the data with ontology are important facts considered in their architecture.

4.0 Formalisms for Representing Ontologies

Depending on the applications, different approaches can be used to represent ontologies. A hierarchy of concepts can be considered an ontology for some applications while others may need much more complicated approaches to represent ontologies. Therefore, representing the intensinal definitions by means is-a relation may be all that is needed for some applications. Some approaches add typed links for representing relations between concepts while others may require more constraint. More requirements implies more complication and more expressive formalism. A suitable formalism for representing ontologies is an important research issue in this domain. *Description Logic* and *Frame logic* are two potential formalisms with existing implemented reasoning systems. They are briefly introduced in the following and the abilities of the reasoning systems are also discussed.

4.1 Description Logic

Description Logic (DL) is potentially one of the means of formalizing and reasoning with intensional definitions. DL is a successor of KL-ONE and is mainly established for knowledge representation. The definition of a family relationships ontology is presented here (by DL in PowerLoom []) to discuss the formalization issues.

The two main formalizing elements in DL are concept definition and relation definition. Concepts are defined by their superconcepts and restrictions on their roles. Definitions (of concepts or relations) are of two types, *primitive* and *defined* (or *non-primitives*). By primitive definitions, one expresses necessary constraints to be satisfied for instances in its extension. Non-primitive definitions are described by necessary and sufficient conditions. Non-primitive definitions can be used when one can give a thorough clear definition of a concept or relation. A DL reasoning system is able to recognize (or classify) individuals under such concepts, implicitly. On the other hand, a DL system can not recognize an individual under a primitive concept unless it is declared explicitly due to the fact that the definition is partial. An example of a primitive concept definition is as following:

```
(defconcept Person (?p Thing)
    :=> (and (exists (?m)
             (and (Person ?m)
                (= (gender ?m) FEMALE)
                (= (mother ?p) ?m)))))
```

It defines a primitive concept “Person” which is a subconcept of “Thing” - “Thing” is a general concept and all concepts are subconcept of “Thing”. It also states that all persons have at least one mother whose “gender” is filled by “FEMALE”. (A DL reasoning system does not conclude that everything with at least one female mother is a person.)

Non-primitive concepts man and woman are defined as following:

```
(defconcept Woman (?p Person)
    :<<=>> (and  (Person ?p)
                (= (gender ?p) FEMALE)))
(defconcept Man (?p Person)
    :<<=>> (and  (Person ?p)
                (= (gender ?p) MALE)))
```

The concept “Woman” is a subconcept of “Person” and its “gender” is filled by “FEMALE”. Therefore, every instance of “Person” whose “gender” role is filled by “FEMALE” will be classified under “Woman” concept in a DL system. By such definition for concept “Woman” one can redefine the concept “Person” as following:

```
(defconcept Person (?p Thing)
    :=> (and (exists (?m)
             (and (Woman ?m)
                (= (mother ?p) ?m)))))
```

Here are examples of primitive and non-primitive relation definitions:

```
(defrelation gender ((?p Person)(?s Gender)))
```

```
(defrelation child ((?p Person) (?c Person))
    :=> (> (age ?p) (age ?c)))
(defrelation son ((?p Person) (?s Man))
    :<<=>> (and  (Person ?p)
                 (Man ?s)
                 (= (child?p) ?s)))
```

Definition of “gender” and “child” relations above are primitive and definition of “son” is non-primitive. That is, if DL reasoning system finds a “child” relation between a Person and a Man it concludes that the “son” relation can be established between them, too.

Originally, DL systems consists of two modules: Terminological Box (TBOX) and Assertion Box (ABOX). TBOX refers to the part that concepts and relations are defined and ABOX is where the individuals are defined[2]. DL reasoning systems focus on inferring subsumption relations between concepts and relations in TBOX (i.e., they can determine where a concept can be located in a specialization hierarchy) as well as recognizing individuals defined in ABOX under a concept definition. This capabilities make DL suitable for reasoning with intensional definitions for interchange of data. As an example consider the following assertion:

```
(assert (= (son SUE) JOHN)
```

That concludes, JOHN and SUE are instance of Person; JOHN is an instance of Man, consequently, its gender is set to MALE; and the child relation will be set between them. On the other hand assertion such as:

```
(assert (= (gender GABRIELLE) FEMALE))
(assert (= (son SUE) GABRIELLE))
```

are incoherent according to our understanding of the above definitions. However, later assertions can cause a reasoning system to conclude that the concepts Man and Woman as well as the individuals MALE and FEMALE are equal.

Consistency checking is important in checking the validity of ontological definitions (in TBOX). Besides, we need to evaluate if the assertions comply with the ontological definitions. Theoretically, DL systems have the ability of finding incoherence, but in practice it depends on the system implementation. As an example, Loom [] with a very expressive language offers the functions to detect incoherence in concept and individual definitions, but it does not detect many examples of incoherence. PowerLoom with an expressive language is able to detect incoherence, however it only reacts by a warning message and does not support any special function for such purpose unlike its ancestor Loom[3].

---

[2] PowerLoom does not have TBOX and ABOX therefore it does not distinguish between concept definitions and assertion. From this point of view it is similar to FLORID. FLORID is a formalism for Frame logic and introduced later in this report.

[3] Since PowerLoom does not distinguish between terminological and assertion definitions therefore it can only detect the incoherence. In case of an incoherence, determining whether terminological definition or the assertion definition is not *correct*, would be a difficult task. It is important to differentiate between the notions of correctness and coherence here. Coherence refers to internal agreement of the definitions, while correctness refers to agreement of definitions with our intension.

NeoClassic is another implementation of a DL reasoning system. While, Neoclassic [] can detect the incoherence in both concept and individual definitions but the Neoclassic implementation of DL is not as expressive as Loom. For instance, the relation (or role) definition in Neoclassic is week in comparison to Loom. The role definition in Neoclassic is attributed to concepts and one cannot define an independent relation with its constraint. Role definition in Neoclassic offers only the type checking and does not offer the possibility of defining specialization hierarchy of relations or any other type of constraint.

4.2 Frame logic

The main purpose of Frame Logic (or FLogic) is to formalize various aspects of object-oriented paradigm [Kifer et al., 1995], though it has been used for knowledge representation, as well. Classes and methods in FL can be used respectively to represent concepts and relations. Here the definition of the same concepts introduced in section 4.1 are presented in FL using the FLORID [] system. Definition of primitive concepts are done as following:

```
person[mother=>woman].
woman::person [gender*->female].
man::person [gender*->male].
```

If an individual is defined as an instance of concept “woman” the system concludes that it is a “person” and it has a role “gender” filled by value “female”. To change the concept “woman” to a non-primitive concept the following rule must be declared:

```
X:woman          :- X:person[gender->female].
```

Consequently, if an object of type “person” has a “gender” method resulting in “female” value, the system concludes that the object is an instance of concept “woman”. The only way to define non-primitive concepts in FL is applying such rules.

Defining the relations can be done by method definitions:

```
person [gender=>sex; child=>>person; son=>>man].
```

The only constraint expressed above is the type checking of the method. To describe that relation “son” is a specialization of child the following rules must be declared:

```
X[child->>Y]     :- X:person[son->>Y:man].
```

By adding the following rule the “son” method will become a non-primitive relation:

```
X[son->>Y]       :- X:person[child->>Y:man].
```

FL does not distinguish between individuals and concepts - i.e., unlike DL, it has nothing such as TBOX and ABOX. That is, an instance of a concept can be considered a concept itself and has its instances. The FL reasoning system performs the same main tasks of deducing implicit knowledge and classifying individuals.

```
book[title=>name].
dictionary::book.
oal_dictionary:dictionary[title*->oxford_advanced_learner].
oxford_advanced_learner:name.
my_dictionary:oal_dictionary.
```

Such definition is not theoretically allowed in DL.

FL is not capable of automatic detection of incoherent definitions unlike DL. In fact, due to the lack of negation in the consequent part of rule definitions, FL reasoning system does not face incoherence. The following definitions:

```
gabrielle : person [gender->female]
sue [son->>gabrielle].
```

will cause FLORID to conclude that “male” and “female” are equal - consequently man and woman as well. Due to the lack of negation there is no way to state that male is not equal to female (are they finally equal or not???). However one can define necessary predicts to find out incoherence. As an example the following rule:

```
type_incoherent(O) :- O[M->>R], O:C, C[M=>>T], not R:T.
```

states that if an objects method results in another object that does not comply with the type definitions of the object class it is considered type-incoherant. This gives more flexibility for defining ontologies and at the same time more responsibility.

4.3 General Features of a Formalism

There are features generally available by formalisms that makes them capable to support representation of ontologies. Here are proper features for ontology representation:

a. Concept definition: Concept definition is a means to define intensional relations of arity one (unary predicate). Concepts can be compared to classes of objects in object-oriented paradigm. Similar to class definitions in concept definitions we need *is-a* relation to define hyponym (and hypernym) or specialization relation. By that one can establish a hierarchical taxonomy of concepts. It is important to note that the comparison to object-oriented paradigm, here, is only for better understanding otherwise there is clear difference between concepts and object classes. For instance, wide street should be defined as a concept in an intensional definition while it may be a numerated value for an attribute street type in a class definition or it can be a class by itself.

b. Relation definition: Relation definition is a means to define intensional relations of arity two or higher. Relations are not merely defined by typed attributes carrying referential keys, Relations in ontologies may be defined independent of concepts. This gives an identity to relations independent from concept definitions (in contrast to roles). For instance, in definition of a brotherhood relation we can state that the relation is established only with a person whose gender is male and that person should have the same parents. (This can help to have a hierarchical taxonomy of relations like taxonomy of concepts and deduce new relations between concepts or instances of concepts which are not explicitly stated). How can one decide to express an intensional relation either by concept definition or by relation definition depends on the domain. But, as a rule of thumb one can say that we tend to express adjectives and adverbs by relation definitions than concept definitions - e.g., above, whiteness, color. [Relations are not rigid. Individuals take part in them or leave them without changing their individuality - we consider them still the same individual.]

c. Role definition: Concepts in an ontology are defined partly by the aggregation of roles. Roles (also called slots) are where a concept establishes a relation to another concept. For instance, “brother” is a role (or slot) for a person where a relation “brotherhood” is established to another male person. Thus, some male persons can play the role of “brother” for some persons - while playing the role of “father” or “enemy” for other persons. The advantage of using roles is that one can specialize a more general relation according to a particular concept in the domain of the relation. For example, while relation “spouse” maps persons to persons, the role spouse defined for man can have a constraint specifying that spouses of (heterosexual) men should be women. (Note that this can be done by more complicated relation definition).

d. Individual (or instance) definition: Instances represent members of the domain by a collection of facts. (Instances can be compared with objects in object-oriented paradigm.) Facts related to every instance is represented by means of its roles. For example, an instance of the concept person can be defined by its roles such as its social security, its name and/or its brother. Instances can be classified under concept definitions (by constraints on its roles and fillers stated for the roles). An instance can play (or fill in) a role for another instance, in other word they can also take part in instances of relations. The difference between instances and concepts is very tiny and subjective.

e. Constraints (or restriction) definition: Restrictions are mostly applied to the roles and their fillers to classify concepts, relations or instances. They may be represented as separate rules in some different implementations, but with the same purpose.

As far as a formalism can provide above-mentioned features it can be used to represent ontological definitions. Formalisms such as ER diagrams, UML diagrams and RDF schema are alternative formalisms for representing ontologies, while there are no reasoning system being able to process such formalisms, at the time this report is written.

5.0 Ontology and Semantic based Integration

An ontology cannot be forced to a community. In fact, the idea of using ontologies is supposed to give the communities the freedom to define their ontology based on their conceptualization. That means we should move towards a common ontology based on ontologies from different systems or communities. It helps where two communities are willing to communicate based on common understanding. A common ontology guarantees the consistency of the understanding of communities from the world - it will not necessarily represent the complete conceptualization of every community committed to the common ontology.

Models[4] of the world are based on conceptualization. Also, it can be considered a projection of the world structures. The modeling process performs this projection and is done according to our interest (Figure 3). As discussed in section 2, the nature of the intensional relations are not known to us and we use axioms in our ontologies to express the relations among them. These axioms can help us to assure consistency of conceptualizations between communities. During any exchange of data from one model to another we need to guarantee the consistency by making sure that the portion of the data integrated to the other model would not be interpreted more than what the original projection to that model would produce. It requires the type of higher level knowledge that formal ontologies can provide by the axioms defining intensional relations.

---

[4] By “model” we refer to a representation of a state of the universe of discourse. It is close to what is called database state in database community. It refers neither to conceptual model (e.g., a UML diagram) nor to the database modeling approaches (e.g., relational or hierarchical) nor to the database description (data model).

It is important to note the two following cases in integration and interoperability:

1. Models are based on the same ontology. In such case our problem is limited to schema integration or finding synonyms and homonyms.

2. Models are based on two different ontologies. This happens when two communities using different ontologies try to interchange or integrate data. In this case the problem is finding a common ontology (or an integrated ontology). It can be done through finding the similarities between concepts defined in two ontologies.

In this section we introduce merged-ontology and show how it can help in solving heterogeneity problems. Merged-ontology can be produced by taking the shared part of underlying ontologies and finding the similarities between them. An ontology is a base to share parts of the conceptualizations and in turn it helps to have the same interpretation of information. It is the only window, two communities can exchange data through.

5.1 Approaches to Use Formal Ontologies

Two possible approaches are introduced here to use ontologies for integration. One is an on-the-fly approach which is similar to approaches introduced by On2broker and SHOE. The second approach is based on generation of a global schema.

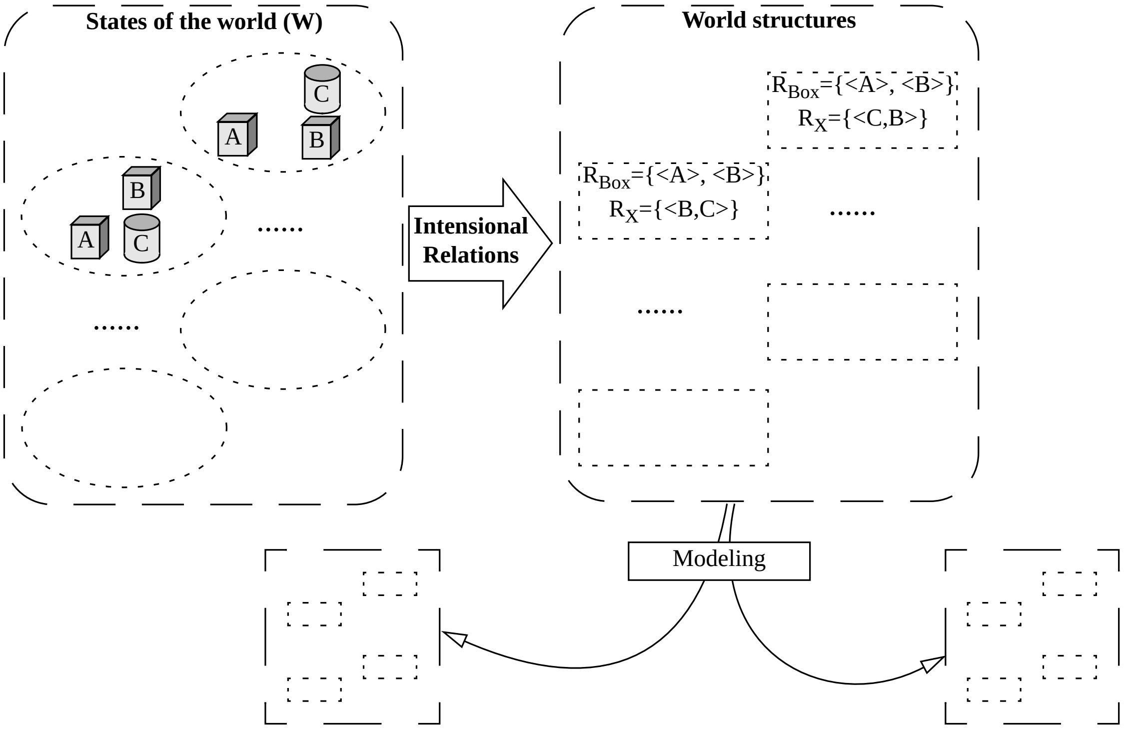

FIGURE 3. Models are projection of world structures build by some constraints according to the requirements.

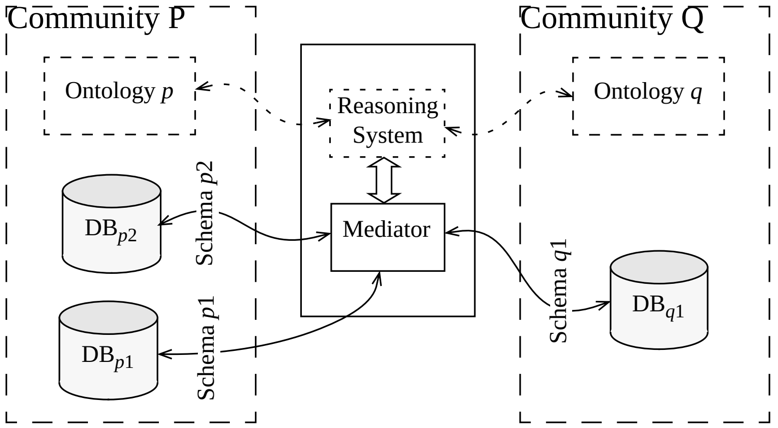

FIGURE 4. On the fly integration - with local queries committing to a domai ontology and no global schema or global query.

In the first approach (as shown in Figure 4) a reasoning system is able to find the similarities between concepts in two ontologies and the mediator will map the corresponding items in two schemas. In this approach the query should commit to the domain ontology and should introduce its domain ontology to the system. Mediator will use a reasoning system to find the mapping between schemas. This is a suitable approach where we have dynamic schemas - such as DTDs in XML data - and when number of data producers change frequently. While, one drawback of this approach is the high processing cost, since for every query the mediator should process respective ontologies and derive required mappings. On the other hand, considering that it is an on the fly approach, a human supervision or semiautomatic approach to find similarities may not be possible therefore it is an error prone approach, as well.

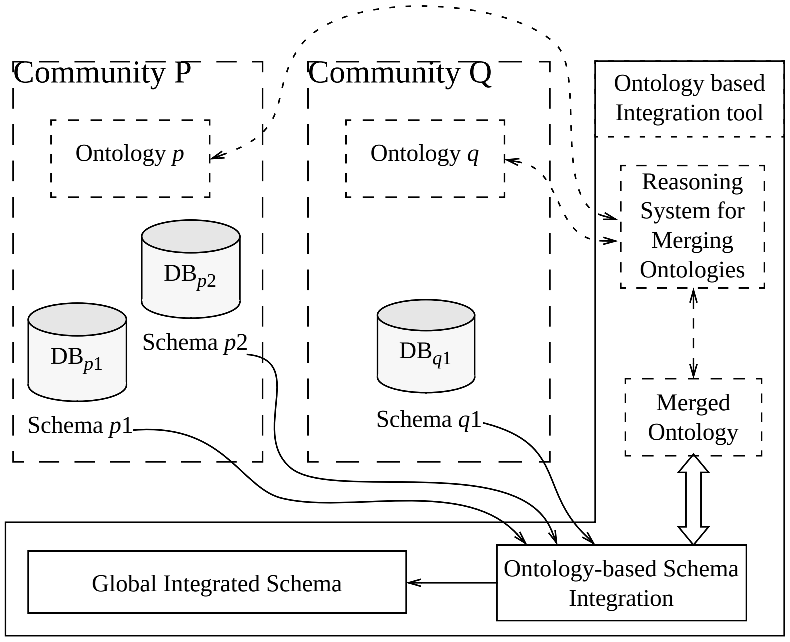

FIGURE 5. Global schema generation based on a common ontology produced by integration of domain ontologies.

The second approach (Figure 5) is appropriate in case of need for a global schema. Therefore, it is suitable where the schemas are not subject to frequent changes. In the proposed approach, a reasoning system is used to generate the merged-ontology[5] derived from domain ontologies. The merged-ontology is produced based on similarities between intensional definitions. Finding similarities can be done by human supervision or a semiautomatic method. Generation of the global schema will be done based on the derived merged-ontology. At this stage merged-ontology will be applied to combine schema items

---

5. Note that we do not use the term integrated ontology to avoid a wrong imprecision that any of the communities would commit the result of the merging process. Thus, we do not try to detect or resolve mismatches between ontologies as in [Visser et al., 1998]

among the component schemas. That is to find all the possible mapping between the generated global schema and all the component schemas. Like, previous approach queries in this architecture should commit to one domain ontology. As the number of underlying databases and the communities increase, number of derived mapping increases, while many of them may not be used by the applications. Thus, a statistical approach or human supervision should be considered to maintain only the applicable mapping.

5.2 Similarity Relations among Ontological Definitions

The main task of the inference engine between ontologies (in both architectures introduced above) is to find *similarities* - or in contrast *differences*. To that end, we establish similarities between terms defined in two formal ontologies. Detection of the similarity relations is based on the axioms specifying the intensional definitions[6] of concepts or relations (represented by ι in the following elaborations). The implications of these similarity relations on involved extensions of intensional relations [Guarino, 1998a] (shown by ε) can be determined. We use the term *concept* to refer to intensional relations with arity 1 and *relation* to refer to intensional relations of arity greater than one. Four levels of similarities between two coherent intensional definitions (i.e., with non-empty extension) can be identified as following[7]:

1. Disjoint definitions: This level has the lowest degree of similarities. Two concepts or relations are disjoint if conjunction of the intensional definitions of the them implies *false*. It follows that the extensions of the two concepts or two relations are disjoint - e.g., narrow street and high way, truck and employee, or sister and father.[8]

   $$((\iota[{}^{p}C_i] \wedge \iota[{}^{q}C_j]) \equiv False) \Rightarrow ({}^{p}C_i \perp {}^{q}C_j)$$

   $$({}^{p}C_i \perp {}^{q}C_j) \Rightarrow (\forall x)\neg(x \in \varepsilon[{}^{p}C_i] \wedge x \in \varepsilon[{}^{q}C_j])$$

2. Overlapping definitions: If the conjunction of two intentional definitions cannot be proven to be *false* (a reasoning system may not necessarily consider it *true*), then they *overlap*. This is, an instance of the definition C_i in ontology *p* may or may not be an instance of the definition C_j in ontology *q*. It depends on the facts stated about the instances and the intensional definition of C_j. It implies that the extension of the definitions intersect - e.g., employee and student, or colleague and sister. (In practice, all concepts are overlap, unless otherwise is proven by intensional definitions.)

   $$(((\iota[{}^{p}C_i] \wedge \iota[{}^{q}C_j]) \equiv \iota[C_k]) \wedge (\iota[C_k] \neq False)) \Rightarrow ({}^{q}C_j \neq {}^{p}C_i)$$

---

[6] Intensional definitions are definition of terms by logical axioms. These logical axioms are estimating every *intensional relation* (defined in [Guarino, 1998a]). For instance, “Faculty” is a conceptual relation and its intensional definition is: ι[Faculty(x)] = Person(x) $\wedge$ ($\exists$y: Course(y) $\wedge$ teaches(x,y))

[7] The case of homonyms is not being considered here due to the fact that all the intensional definitions for a particular community must be identified by their respective terms and known to and agreed by all members of one community - otherwise, it can cause confusion in the community. In case of intercommunity, all the intensional definitions and their respective terms ported outside the community should be uniquely identified according to their respective community [Visser and Cui, 1998] - i.e., terms coming from different communities are considered referring to different concept, unless, otherwise can be proved.

[8] When combination of the intensional definitions of disjoint definitions is equal to the generalized definition of the two definition, we say they partition the generalized definition. In our example man and woman partition person, that is, a person belongs to one (and only one) of the two disjoint subconcepts.

$$({}^pC_i \neq {}^qC_j) \Rightarrow (\exists x)(x \in \varepsilon[{}^pC_i] \wedge x \in \varepsilon[{}^qC_j])$$

3. Specialized definitions (subconcepts or subrelation): If the intentional definition of C_j is an implication of the intensional definition of C_i, then C_i is a specialization of C_j. Hence, if a definition C_i in ontology *p* is a specialization (or hyponym) of C_j in ontology *q* then every instance of the definition C_i is an instance of C_j. This implies that the extensions are in a subset relation. For instance, “man” is a subconcept of “person” and “wife” is a subrelation of “spouse”. The specialization similarity is a partially ordered relation.

$$((\iota[{}^pC_i] \wedge \iota[{}^qC_j]) \equiv \iota[{}^pC_i]) \Rightarrow ({}^pC_i \leq {}^qC_j)$$

$$({}^pC_i \leq {}^qC_j) \Rightarrow (\forall x)(x \in \varepsilon[{}^pC_i] \Rightarrow x \in \varepsilon[{}^qC_j])$$

4. Equal definitions: This level has the highest degree of similarity. If the intensional definition of the two intensional definitions are equivalent, then the defined concepts are equal. Therefore, every instance of the C_i under ontology *p* would be an instance of C_j under ontology *q* and vice versa. According to the above definition, if two concepts or relations are equal, each of them specializes the other one, respectively. Furthermore, the corresponding extensions are equal. For instance, “vehicle” and “transportation facility” are equal if they have the same intensional definition.

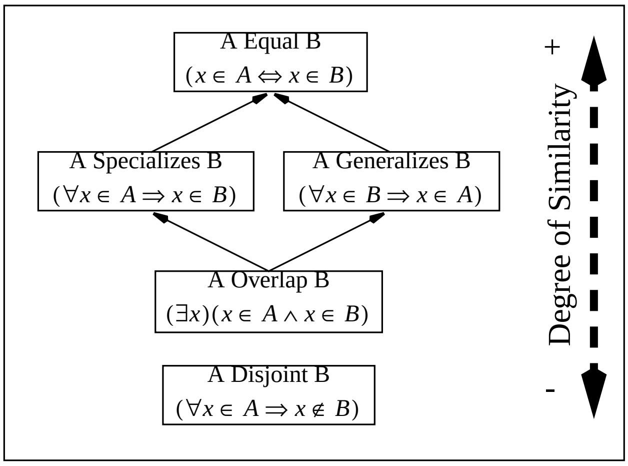

FIGURE 6. Similarity relations among ontological definitions

$$(\iota[{}^pC_i] \equiv \iota[{}^qC_j]) \Rightarrow ({}^pC_i = {}^qC_j)$$

$$({}^pC_i = {}^qC_j) \Rightarrow (\forall x)(x \in \varepsilon[{}^pC_i] \Leftrightarrow x \in \varepsilon[{}^qC_j])$$

Above similarity relations can be driven by the 4-intersection approach used to derive topological relations between spatial objects in [Egenhofer and Herring, 1991]. One can use logical *and* instead of intersection; *true* and *false* instead of *empty* and *non-empty* sets; and intensional definition and its negation instead of interior and boundary. Crisp logic and crisp set theory results in the above levels of similarities, using other approches such as multivalued logic might give rise to more levels of similatities between intensional definitions.

Deriving similarities between ontologies requires common references in two ontologies and a reasoning system (heuristic) for matching. The common references can be provided by a higher level ontology such as Ontolingua [Farquhar et al., 1997] or by a thesauruses such as WordNet (as suggested in the KRAFT project). Finding similarities can also be done by experts familiar with both communities or by a hybrid semiautomatic method. Another approach could be keeping part of similarity relations in a repository (as in OBSERVER) and trying to infer new relations from the relations stored in repository.

5.3 Merging Ontologies and Schema Integration

Similarity relations introduced above can help to merge two ontologies and to produce integrated global schema afterwards. The disjoint definitions are not discussed at this phase and

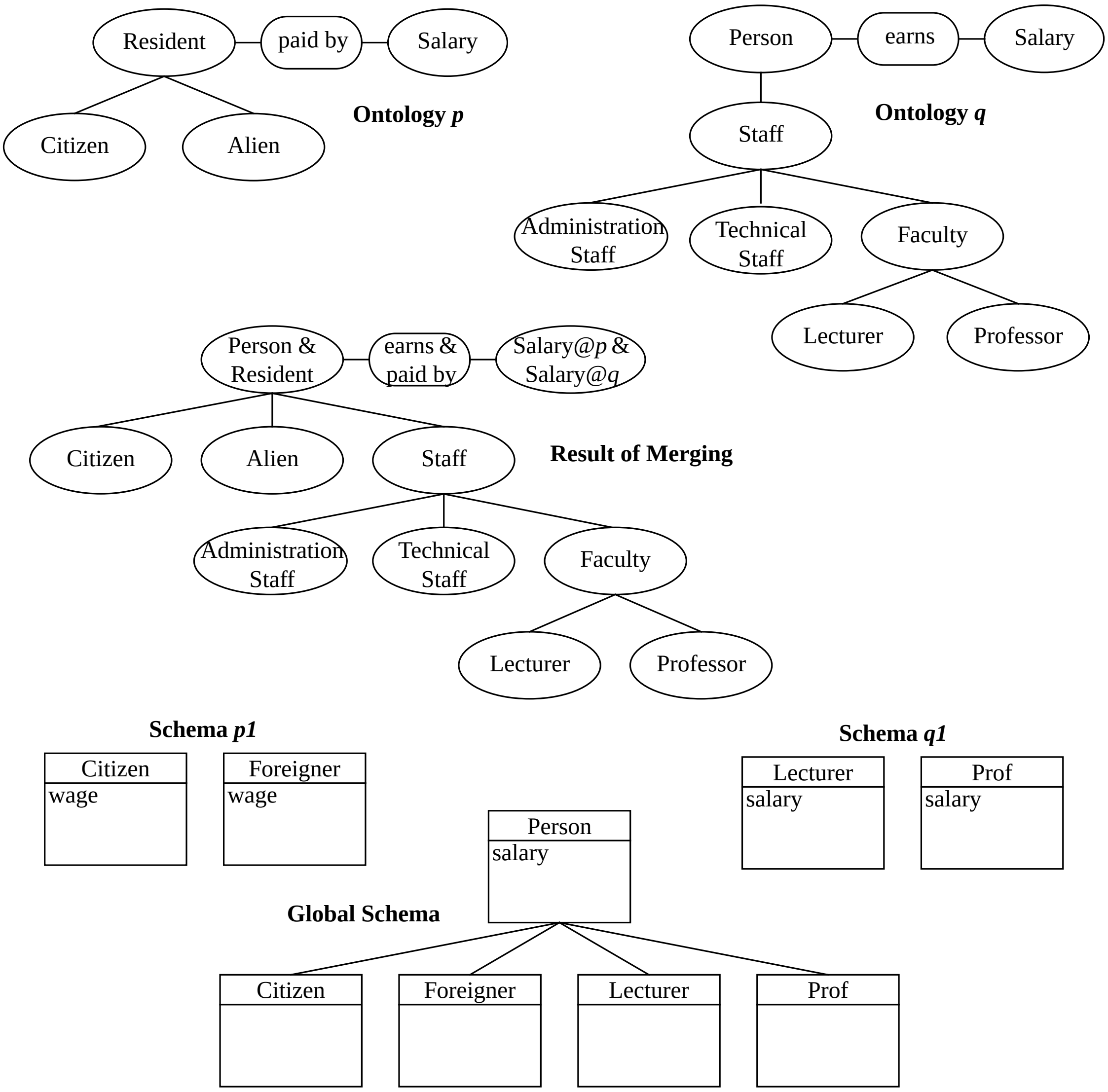

FIGURE 7. Result of finding two equal concepts and creating a class Person in global schema.

nothing will be done about them in this phase. We take all the intensional definitions in the respective communities for merging process and explicitly establish similarity relations as explained in the following.

1. If two definitions are equal, the result of merging is a unique intensional definition which is referred to by both original terms. That is, different terms in the local schema definitions can refer to the same concept - synonym terms such as “Person” and “Resident” in Figure 7[9] (we use *&* sign to show them in the figures).
2. If an intensional definition C_i specializes C_j then the subconcept or subrelation similarity will be explicitly established between them (e.g., “Student” and “Person” in Figure 8). Therefore, C_i will inherit from C_j. If C_i specializes any subconcept or subrelation of C_j then

---

[9] For sake of simplicity in figures 2, 3 and 4 we only illustrate parts of the taxonomy trees of ontologies and part of the schemas. Dashed lines show the similarity relations established during merging ontologies.

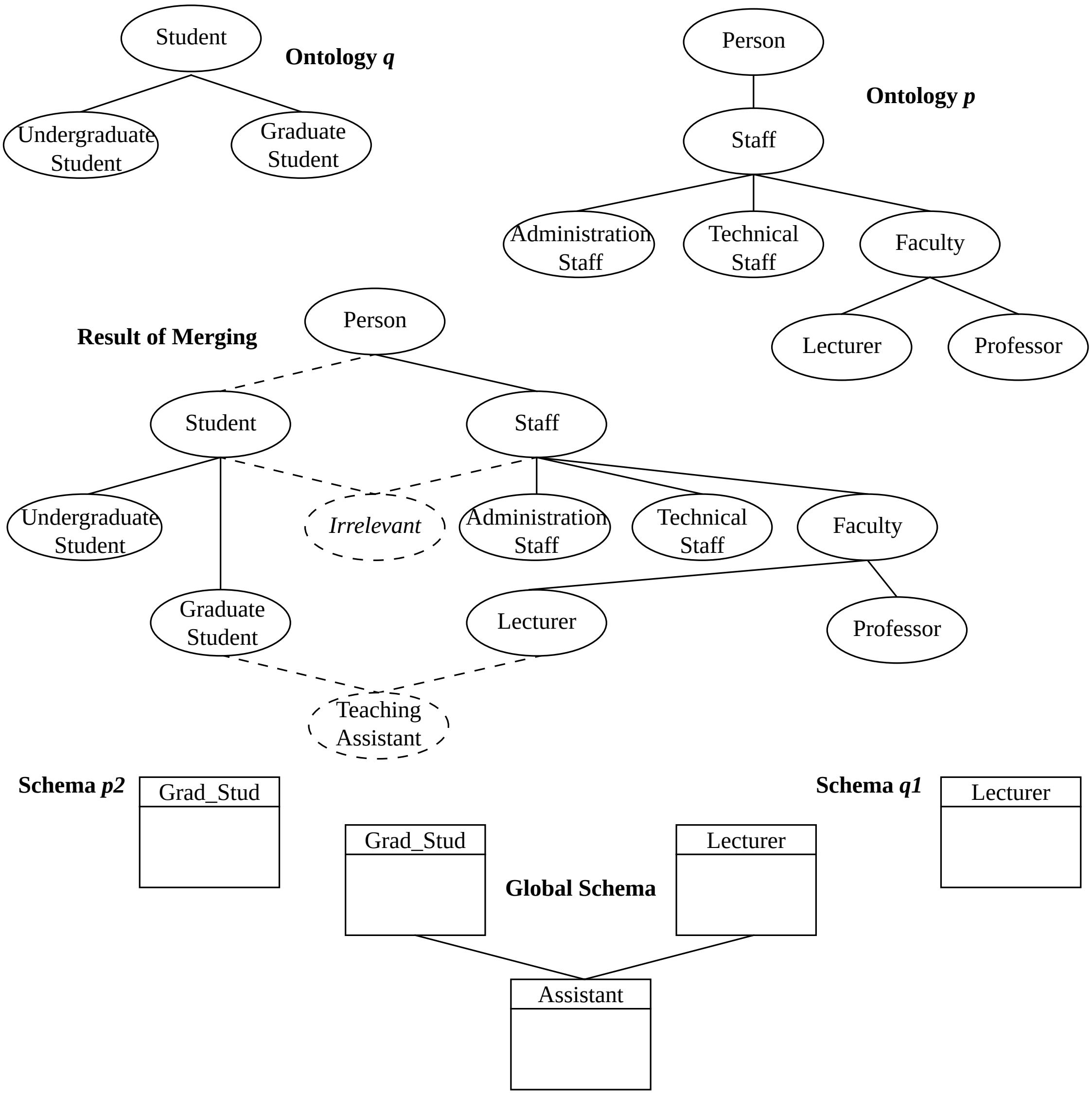

FIGURE 8. Establishing similarity relations, creating new concept and new class based on the underlaying ontologies.

such a relation will not be explicitly stored, since it can be deduced based on the transitivity of specialization (e.g., “Graduate Student” and “Person” in Figure 8).

3. If a definition C_i overlaps with C_j then an additional new concept or relation will be declared as the conjunction of the two intentional definitions. The new concept will inherit from both existing ones and assigning a term to the new concept or relation can be done automatically or by an expert. Deciding if such a concept is relevant or not should be done by an expert, as well (e.g., concepts “Staff” and “Student” or “Lecturer” and “Graduate student” in Figure 8). An important issue to consider is to find and establish relations each concept in merged-ontology and every other concept.

For every of the above mentioned similarities, the schema integrator may face any of the following affair between schema items. Any of the following can happen when integrating schemas in the same community and based on one ontology [Kim et al., 1993].

1. Class vs. Class

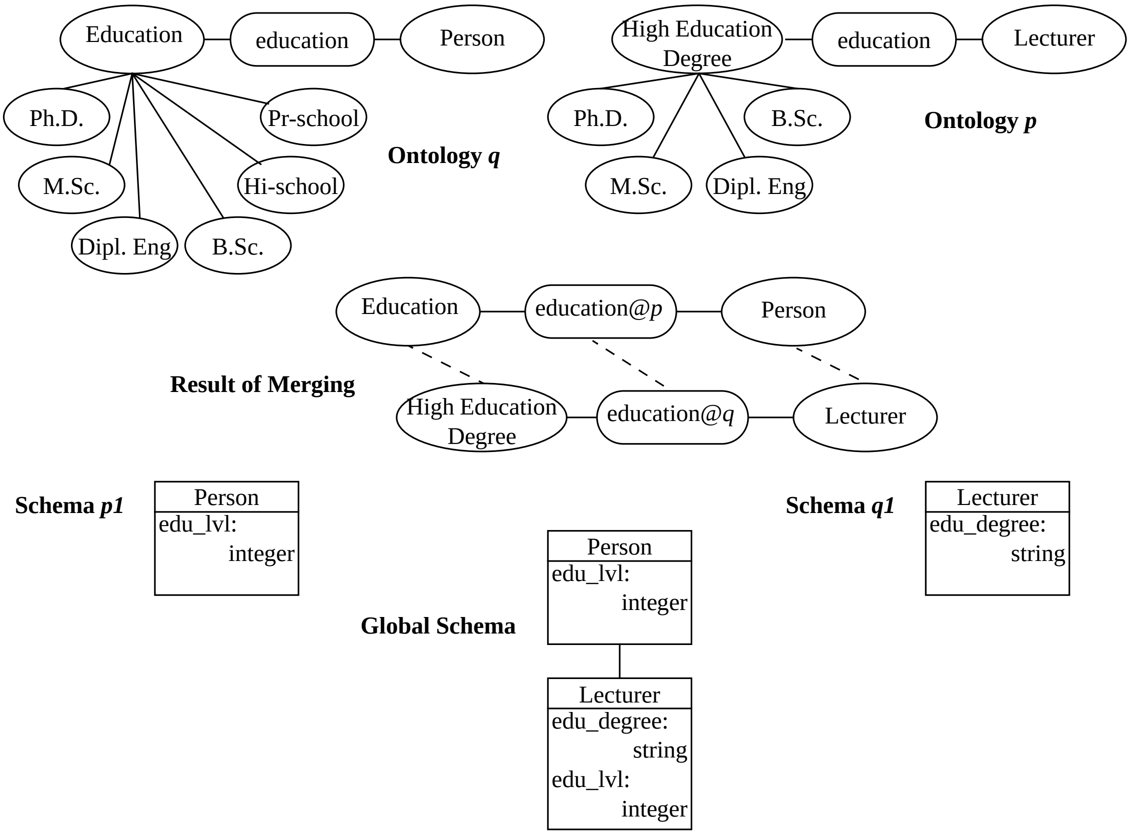

FIGURE 9. Example of detecting a subrelation similarity.

- Whenever two classes in two schema definitions are referring to the same concept, then one global class definition will represent the two classes in the component schemas. The global class definition will subsume the two local class definitions; see also below for the definition of global class attributes. We need a criterion to realize that two objects in the underlying databases representing the same individual [Guarino and Welty, 2000b]. This *identification criterion* must be present in both local class definitions, such as social security number in our example. It may not be the primary key in one or both systems, though. Afterwards, a mapping between the two classes and the integrated class in the global schema can be defined.
- If two classes are referring to two concepts in a (direct or indirect) specialization similarity, then the classes should be in a (direct or indirect) specialization relation in the global schema (e.g., classes “Person” and “Lecturer” in Figure 7). The specialized class will be defined by the union of the definition of both classes - union of class definition are discussed further in the part related to attributes. In addition to the identification criterion, we need *classification criteria* to map instances of the superclass (“Person” in one database) to instances of the subclass (“Lecturer” in the other database). A reasoning system can find such criteria by referring to the intensional definitions - for non-primitive definitions.
- If two classes refer to two overlapping concepts, then an overlapping class is added to the global schema based on conjunction concept. The new class will inheriting from both classes. Classifying objects under the new class in the global schema is done again based on the intensional definition of the conjunction concept (e.g., class “Assistant” based on concept “Teaching Assistant” in Figure 8).

- If two classes refer to two overlapping or disjoint concepts, while the corresponding concepts have a common superconcept, a class based on the common superconcept is defined in the global schema, such as “Lecturer” and “Citizen” in Figure 7.

For union of two class definitions we need the union of the class definitions based on attributes. We consider all attributes to be based on relations in ontology definitions. This concludes that the domain of the respective relations for attributes are already in a similarity relation when we start to integrate them. We classify the affairs between attributes according to the range of attributes. A specific case is discussed here when range of an attribute is not filled by individuals under a class but by subclasses of a class.

2. Attribute vs. Attribute

- In the first case, attributes in two classes are referring to the same relation (e.g., relation “earns & paid by” in Figure 7). One attribute will appear in the global class definition representing both attributes (e.g., attribute “salary” and “wage” Figure 7). If attribute types are different (e.g., integer and real) or use different units, then the mapping of data needs to be done with value conversion - data conversion is out of the scope of this report.
- Second, two attributes might refer to two relations in a specialization similarity - e.g. education in Figure 9, “Example of detecting a subrelation similarity.,” on page 21 or “sibling” and “sister” attributes. Note that to show the generality of the approach, the range of the relation (i.e., “education”) is not given by individuals under a concept (i.e., “Education Degree” or “High Education Degree”) but by subconcepts. If the range of an attribute is formed by a subconcept (a set of individuals or a range of values), the mapping will be more complicated. For example, attribute “width” with two possible values `wide` and `narrow`, or “edu-lvl” and “edu-degree” in Figure 9, “Example of detecting a subrelation similarity.,” on page 21. Classifying the subconcepts may vary in different communities in terms of covered ranges of values or the granularity of classification. In Figure 9, “Example of detecting a subrelation similarity.,” on page 21 one can consider the case that “Education Degree” has three subconcepts: Postgraduate, Graduate and School, while “High Education Degree” is still a subclass of “Education Degree”. Although, “Education Degree” has a finer granularity, the approach to define the attribute in the global class remains the same. The attribute related to the general relation must be kept in the global class definition. Mapping from specialized attribute (e.g., “sister” or “edu_degree”) to the general attribute (e.g., “sibling” or “edu_lvl”) can cause information loss, and the reverse mapping from specialized attribute to general attribute may cause imprecision. Information loss does not cause problem for the user since she does not require finer data. While imprecision requires a mechanism either to find the right mapping by means of the reasoning system and ontological definitions (depending on the state of the individual) or to inform the user of the imprecise data. One can keep both general and specialized attributes in the global class and perform all the mappings at the global schema level—*not* during the mapping of data between the global database and local databases.
- Two attributes are referring to two relations in specialization similarity - e.g. sibling and sister attributes or two attributes bounded-at and ends-at for arcs bounded at starting and ending nodes. The attribute related to the more general relation must be kept in the class definition. Mapping from general attribute (in global schema) to the specialized one will cause information loss and the reverse mapping from specialized attribute to general attribute causes imprecision. Information loss does not cause problem for the user since s/

he does not require finer data. While imprecision requires a mechanism either to find the right mapping by means of the reasoning system and intensional definitions or to inform the user of the imprecise data. One can keep both general and specialized attributes in global schema and do all the mappings in the global schema - not during the mapping of data to/from global schema.

- Two attributes are referring to two overlapping relations. To avoid any information loss at the global schema both attributes should be kept in the class definition while during the mapping for every object at least one of the attributes should be filled and if the reasoning system is able to derive required mapping the other can be filled according to the mapping. Just like last case a mechanism is required to inform the user of the low quality of the data in case of imprecision.
- If the range of an attribute is filled by a subconcept (a set of individuals or a range of values), the mapping will be more complicated. For example attribute width with two possible values wide and narrow, or “edu-lvl” and “edu-degree” in Figure 9. Classifying the subconcepts may vary in different communities in term of covered range of values or granularity of classification. In Figure 9 one can consider the case that education-level has three subconcepts: postgraduate, graduate and school levels. In this case “High Education Degree” is still a subclass of “Education Level”. Although, “Education Level” has a finer granularity.
- Finally, two attributes can refer to two overlapping relations. To avoid any information loss at the global schema both attributes should be kept in the class definition and a mechanism is required to inform the user of the low quality of the data in case of imprecision. However, we did not find an example of such case.

6.0 Conclusion and Further Work

In this paper we address terminological semantic heterogeneity in creating a global schema. We show how ontologies can be used to solve such problem and create global schema for federated database systems based on ontologies. We also classify semantic problems arising during creation of global schemas. This classification is based on the possible solutions offered by ontology definitions and the capabilities of reasoning systems.

In our further work we are planning to address the following problems:

- We need a clear definition of commitment of an schema definition to an ontology. By that we are referring to an approach to validate commitment of a schema to its ontology. So far, we consider using a link between the terms in the schema definitions and the definitions in the ontology while skipping the fact that such commitment has consequences on the schema definitions which in turn are result of the conceptual modeling approches. The relations between schema definitions and intensional definitions should be discussed and investigated.
- Methods are considered parametric attributes in object-oriented paradigm, therefore, we did not discuss them as different issue from attributes. However, one may consider the semantics of methods different from that of attributes. In this case, semantics of methods can be represented by verbs and actions ontologies.

- The IS-A relation is used to build taxonomy hierarchy in our ontologies. It is the only predefined relation in our ontologies - i.e., one does not have to define it by defining a relation. While aggregation relations has to be explicitly defined by means of relation definitions and logical axioms. Considering aggregation relations as a predefined relation can help us to simplify the process of building ontologies and improve the integration process - that is taking properties and problems of mereology into account.
- Building an ontology is a pragmatic and very essential problem for this approach. We need well defined ontologies for its success. Therefore, finding a way of defining and evaluating ontologies is of high priority.

7.0 References

Benjamins, V. R. and Fensel, D. (1998). The ontological engineering initiative (KA)^2. In Guarino, N., editor, *Formal Ontology in Information Systems*, pages 287–301. IOS Press.

Bergamaschi, S., Castano, S., di Vimercati, S. D. C., Montanari, S., and Vincini, M. (1998). An intelligent approach to information integration. In Guarino, N., editor, *Formal Ontology in Information Systems*, pages 253–267. IOS Press.

Bishr, Y. A., Pundt, H., Kuhn, W., and Radwan, M. (1999). Probing the concept of information communities - a first step toward semantic interoperability. In Goodchild, M., Egenhofer, M., Fegeas, R., and Kottman, C., editors, *Interoperating Geographic Information Systems*, pages 55–69. Kluwer Academic.

Egenhofer, M. J. and Herring, J. R. (1991). Categorizing binary topological relations between regions, lines and points in geographic databases. Technical report, Department of Surveying Engineering, University of Maine, Orono.

Elmasri, R. and Navathe, S. B. (2000). *Fundamentals of Database Systems*. Addison-Wesley, third edition.

Farquhar, A., Fikes, R., and Rice, J. (1997). The ONTOLINGUA server: a tool for collaborative ontology construction. *International Journal of Human-Computer Studies*, 46:707–727. ftp://ftp.ksl.stanford.edu/pub/KSL_Report/KSL_96_26.ps.

Garcia-Solaco, M., Saltor, F., and Castellanos, M. (1996). Semantic heterogeneity in multidatabase systems. In Bukhres, O. A. and Elmagarmid, A. K., editors, *Object-oriented Multidatabase Systems: A Solution for Advanced Applications*, chapter 5, pages 129–202. Printice-Hall.

Goh, C. H., Bressan, S., Madnick, S., and Siegel, M. (1999). Context interchange: New features and formalisms for the intelligent integration of information. *ACM Transaction on Information Systems*, 17(3):270–290.

Grefen, P. W. P. J. (1992). *Integrity Control in Parallel Database Systems*. Ph.D. thesis, University of Twente, The Netherlands.

Gruber, T. R. (1993). Towards principle for the design of ontology used for knowledge sharing. In Guarino, N. and Poli, R., editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation, International Workshop on Ontology*. Kluwer Academic.

Guarino, N. (1998a). Formal ontology and information systems. In Guarino, N., editor, *Formal Ontology in Information Systems, Proceedings of FOIS'98*, pages 3–17, Trento, Italy. IOS Press, Amsterdam.

Guarino, N., editor (1998b). *Formal Ontology in Information Systems*. IOS Press, Amsterdam.

Guarino, N. (1999). The role of identity condition in ontology design. In Benjamins, V., Chandrasekaran, B., Gomez-Perez, A., Guarino, N., and Uschold, M., editors, *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5)*, Sweden.

Guarino, N. and Welty, C. (2000a). Identity, unity, and individuality: Towards a formal toolkit for ontological analysis. In Horn, W., editor, *Proceedings of ECAI-2000 The European Conference on Artificial Intelligence*, Amsterdam. IOS Press.

Guarino, N. and Welty, C. (2000b). Ontological analysis of taxonomic relationships. In Laender, A. H. F., Liddle, S. W., and Storey, V. C., editors, *Conceptual Modeling - ER 2000, 19th International Conference Proceedings, Lecture Notes in Computer Science 1920*, pages 210–224. Springr Verlag.

Heflin, J. and Hendler, J. (2000). Semantic interoperability on the web. In *Extreme Markup Languages 2000*. http://www.cs.umd.edu/projects/plus/SHOE/pubs/extreme2000.pdf.

Jones, D. (1998). Developing shared ontologies in multi-agent systems. In *ECAI'98 Workshop on Intelligent Information Integration*, Brighton, U.K.

Kifer, M., Lausen, G., and Wu, J. (1995). Logical foundations of object-oriented and frame-based languages. *Journal of the (ACM) Association for Computing Machinery*, 42(4):741–843.

Kim, W., Choi, I., Gala, S., and Scheevel, M. (1993). On resolving schematic heterogeneity in multidatabase systems. *Distributed and Parallel Databases*, 1(3):251–277.

Kim, W. and Seo, J. (1991). Classifying schematic and data heterogeneity in multidatabase systems. *IEEE Computer*, 24(12):12–18.

Kottman, C. (1999). *Semantics and Information Communities, The OpenGIS Abstract Specification Topic 14*. OpenGIS Consortium, 35 Main Street, Suite5, Wayland, MA 01778, version 4 edition.

Mena, E., Kashyap, V., Illarramendi, A., and Sheth, A. (1998). Domain specific ontologies for semantic information brokering on the global information infrastructure. In Guarino, N., editor, *Formal Ontology in Information Systems*. IOS press.

Sowa, J. F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks/Cole - Thomson Learning.

Staudt, M., Vaduva, A., and Vetterli, T. (1999). Metadata management and data warehousing. Technical report, Institute of Informatics, University of Zurich., ftp://ftp.ifi.unizh.ch/pub/techreports/TR-99/ifi-99.04.pdf.

Visser, P., Jones, D., Beer, M., Bench-Capon, T., Diaz, B., and Shave, M. (1999). Resolving ontological heterogeneity in the KRAFT project. In *10th International Conference and Workshop on Database and Expert Systems Applications DEXA'99*. University of Florence, Italy.

Visser, P. R. S. and Cui, Z. (1998). Heterogeneous ontology structures for distributed architectures. In *ECAI-98 Workshop on Applications of Ontologies and Problem-solving Methods*, pages 112–119.

Visser, P. R. S., Jones, D. M., Bench-Capon, T. J. M., and Shave, M. J. R. (1998). Assessing heterogeneity by classifying ontology mismatches. In Guarino, N., editor, *Formal Ontology in Information Systems, Proceedings of FOIS'98*, pages 148–162, Trento, Italy. IOS Press.