Resolving Semantic Heterogeneity in Schema Integration: an Ontology Based Approach

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

Abstract — Interoperability and integration of data sources are becoming ever more important issues as both, the amount of data and the number of data producers are growing. Interoperability not only has to resolve the differences in data structures, it also has to deal with semantic heterogeneity. *Semantics* refer to the meaning of data in contrast to syntax, which only defines the structure of the schema items (e.g., classes and attributes). We focus on the part of semantics related to the meanings of the terms used as identifiers in schema definitions. This paper presents an approach to integrate schemas from different communities, where each such community is using its own ontology. The approach is based on merging ontologies based on similarity relations among concepts of different ontologies. We present formal definitions of similarity relations based on intensional definitions and conclude the extensional consequences. The process of merging ontologies based on the detected similarity relations is discussed. The merged ontology is finally used to derive an integrated schema. The resulting schema can be used as the global schema in a federated database system.

Categories & Descriptors — D.2.12 [Software Engineering]: Interoperability - *data mapping*. H.2.5 [Database Management]: Heterogeneous Databases. I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods.

General Terms — Design.

Keywords — Semantic Heterogeneity, Formal Ontologies, Database Integration.

1. Introduction

In many domains, the number of data providers and amount of available data is increasing tremendously. However, users usually require an integrated view of the data available from heterogeneous data sources. Therefore, integration issues are attracting ever more attention. Data integration refers to combining data in such a way that a homogeneous and uniform view is presented to users.

We distinguish two types of heterogeneity here: data heterogeneity and semantic heterogeneity. Data heterogeneity refers to differences among local definitions, such as attribute types, formats, or precision. These differences can be easily resolved. Semantic heterogeneity refers to

---

* The work of Farshad Hakimpour is funded by Swiss National Science Foundation (Project Number: 2100-053995).

FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA.

differences or similarities in the meaning of local data. For example, two schema elements in two local data sources can have the same intended meaning, but different names. Thus, during integration, it should be realized that these two elements actually refer to the same concept. Alternatively, two schema elements in two data sources might be named identically, while their intended meanings are incompatible. Hence, these elements should be treated as different things during integration.

As a consequence, adequate and meaningful data integration relies on the detection of discrepancies and similarities between schema elements. Thus, *semantics* of data must be taken into account during integration. Semantics is the interpretation people attribute to data (i.e., relating data to what they represent) according to their understanding of the world. Different interpretations of data causes semantic heterogeneity. In the database domain, semantics refers to the meaning of schema elements (e.g., classes, attributes and methods). Often, it is used in contrast with syntax, by which we refer to the definition of the structure of schema elements.

Schemas are definitions that specify the structure of data and are the result of a database design phase. In data integration, each local database provides a description of the data it is willing to export (the local schema). The aim of the integration process is the development of a global schema which integrates and subsumes the local schema in such a way that (global) users are provided with a uniform and correct view of the global database. The most urgent problem in this respect is semantic heterogeneity—if it is not detected and resolved, the usage of integrated data leads to invalid results. Even worse, since users do not know about semantic heterogeneity in the data they access, they do not have a chance to realize the invalid results.

Relying on common sense is a critical source of semantic heterogeneity and explicit definition of terms used in schema definitions is a solution to this problem. Explicit and formal definition of semantics of the terms guided many researchers to apply *formal ontologies* [9] as a potential solution of semantic heterogeneity. A formal ontology consists of *logical axioms* that convey the meaning of terms for a particular *community* [3, 17]. Consensus on ontological definitions among members of a community is an important difference between ontologies and conceptual models [5]. While conceptual models are application-dependent, ontologies are only based on people's understanding. Formal ontology is considered more than schema definitions in databases. Schemas are mainly concerned with organizing data in databases, formal ontologies are concerned merely with the understanding of the members of the community and helps to reduce ambiguity in communication. It is important to note that schema definitions are based on the ontology definition and vise versa, they convey part of the knowledge about ontology of a community.

In this paper, we present an approach that indeed uses formal ontologies to derive (global) schemas. Ontologies pertaining to local schemas are checked for similarities; this knowledge about data semantics is then used to resolve semantic heterogeneity in the global schema.

The remainder of this paper is structured as follows. Section 2 briefly introduces relevant related work. Section 3 presents an overview of the architecture proposed to use ontologies. We also briefly discuss the whole process and its characteristics. In section 4, we discuss our interpretation of semantic similarity based on definitions in ontologies. Then, we show how to merge two ontologies based on similarity relations. Section 5 discusses the integration of schemas to produce the global schema for a federated database system. The integration process is based on merging ontologies to avoid semantic heterogeneity. In section 5 we draw the conclusion and our further work.

2. Related Work

Prior related research can be considered from three different points of view. One perspective is related to *building ontologies*. This aspect of research discusses, e.g., "how to build a taxonomy tree" or "how good an ontology conforms to the conceptualization of a community". Building a proper ontology in terms of its explication (i.e., how an ontology reveals implicit assumptions) and its accordance with the conceptualization of the community is an important research issue. Due to lack of space we do not discuss this aspect any further one can refer to [1, 10, 11, 21] for more information.

Another perspective considers the representation of ontologies and reasoning based on ontologies. Topics such as "how to represent an ontological definition" or "what features are important in representing ontologies for reasoning" are addressed here. OBSERVER [18] is a project using ontologies to allow queries against heterogeneous sources. It applies ontologies to replaces terms in user queries with suitable terms in target ontologies. OBSERVER uses Description Logic for representing and processing ontologies. Projects in the framework of Semantic Web such as SHOE [13] and Ontobroker [1] also use ontologies to improve the searching abilities on the Web. Both systems are using their own extension to HTML tags and use logical reasoning based on ontological definitions. In SHOE, ontologies are created as a taxonomy hierarchy and Tags refer to the concepts and relations (respectively, known as categories and relationships in SHOE). OntoBroker also keeps a taxonomy hierarchy by means of IS-A relations and represents attributes in their ontological definitions. As a result of using a formalism similar to Frame Logic, the only means to define relations in Ontobroker are rule definitions.

The last perspective concerns semantic integration of database schemas. This topic is closer to the work presented in this paper and addresses issues such as "how formal ontologies can help to solve heterogeneity problems", "what kind of heterogeneity problem can it solve", "how do we relate formal ontologies with schemas" or "how do formal ontologies interact with existing system architectures".

KRAFT uses shared ontologies [14] as a basis for mapping between ontology definitions and communication between agents. In [21], shared ontologies are "*chosen to make shared ontology as expressive as the 'union' of the ontologies*". However, the definition of the union of ontologies and its similarities or differences with a shared ontology is not stated. KRAFT detects a set of ontology mismatches [20] and establishes the mapping between a shared ontology and local ontologies.

The COIN [8] project presents a suitable architecture for semantic interoperability. The role of the Domain Model in the COIN-architecture can be compared to that of an ontology. However, the Domain Model is more close to a conceptual model. As an example, in [8] "money amount" is considered a subtype of "semantic number" and "currency type" is a subtype of "semantic string" while number or strings are only primitive types for *representing* values. According to our understanding of ontology, an ontology is based on the conceptualization of people in a community. Therefore, "money amount" is, say, an amount or a quantity, but not number. Treating "semantic number" as a supertype is the result of influence of an application domain (the argument of [10] applies here).

Bergamaschi et al. introduced an approach to integrate schemas based on a thesaurus [2]. They extract similarity relations from schema definitions of component databases, not directly from an ontology. The approach is a semiautomatic relation extraction based on schema definitions and needs supervision of an expert. Based on the extracted relations they introduce an algorithm to integrate schema definitions into a global homogeneous schema. The Inter-Ontology Relationships Manager module in OBSERVER [18] maintains the same relations between ontological definitions as presented by Bergamaschi et al. in [2]. By means of inter-ontology relations, OBSERVER replaces terms in user queries with suitable terms in target ontologies.

The work of Kim et al. [15, 16] is a comprehensive study for the classification of schema heterogeneities. Also, they present solutions for several types of schema heterogeneity in RDBs and OODBs. Another comprehensive work in this area is presented in [7]. Both works address problems of schema heterogeneity but do not distinguish between the schematic and semantic issues while we focus on the semantic problems here.

3. Formal Ontologies and Information Integration

There are mainly two trends for using ontologies in resolving semantic heterogeneity. One uses ontologies for translating queries, or their results (as in SHOE, On2Broker or OBSERVER). This approach is suitable whenever schemas are subject to frequent changes (such as DTDs in XML data), when many data sources are involved, or the number of involved data sources changes frequently (such as data sources in the Internet). One drawback of this approach is the high processing cost, since for every query, ontologies must be processed to derive required mappings. On the other hand, human supervision to validate extracted similarities is not possible, due to the need for immediate action. Lack of human supervision makes this approach less reliable — e.g., ontology based search engines.

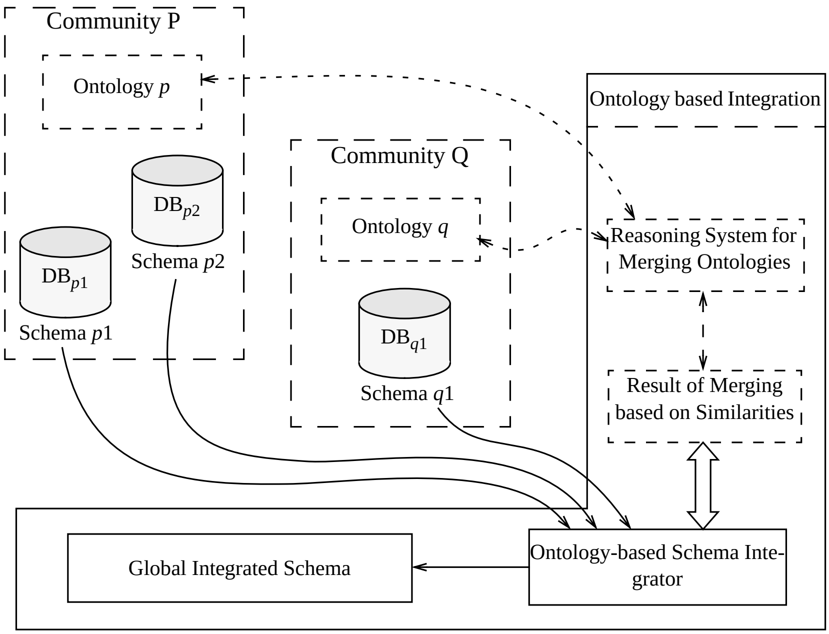

Figure 1. Global schema generation based on a common ontology produced by integration of ontologies.

The second trend uses ontologies for the generation of global schemas [2, 8]. It is suitable whenever the schemas are not subject to frequent changes. In this approach, database schemas commit to the ontology of a community. This is done by relating every term in the schema definitions to a definition in the ontology of the community. In Figure 1 two databases DB_{p1} and DB_{p2} commit to an ontology *p* by referring to the terms defined in the ontology *p*. Such relation can be established either by hard links or by using the same terms as they are defined in the ontology.

In order to find out whether (and how) elements from different schemas are related, we use similarity relations. Detection of similarity is based on intensional definitions of terms represented in a logical language (Description Logic). A reasoning system (such as PowerLoom) can merge ontologies[1] used by the involved communities. Human supervisors and a semiautomatic method cooperate to find these similarities.

Generation of a global schema will be done by a schema integrator based on the result of merging ontologies. Thus, we do not only suggest a global schema, but also try to find all the possible meaningful mappings between the generated global schema and the component schemas. For instance, the schema integrator suggests a class in the global schema as well as the mapping of this class and its attributes into one or more classes in the underlying local schemas. As the number of underlying databases and the communities increase, the number of derived mappings increases, even though many of them may not be used by the applications. A statistical analysis or human supervision to maintain only the valid mappings and schema items can help to overcome this disadvantage.

4. Merging Ontologies by Means of Similarity Relations

Merging ontologies is based on finding *similarities* or *differences* between intensional definitions. To that end, we establish *similarity relations* between terms defined in two ontologies. Detection of the similarity relations is based on the intensional definitions[2] (represented by ι in the following elaborations). The implications of these similarity relations on involved extensions of intensional relations [9] (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 1. Four levels of similarities between two coherent intensional definitions (i.e., with non-empty extension) can be identified[3]:

1. *Disjoint definitions*: This level has the lowest degree of similarity. Two concepts or relations are disjoint if the conjunction of their intentional definitions implies *false*. It follows that the extensions of the concepts (or relations) are disjoint—e.g., narrow street and high way, truck and employee, or sister and father.

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

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

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.

   $$(((\iota[{}^pC_i] \wedge \iota[{}^qC_j]) \equiv \iota[C_k]) \wedge \neg(\iota[C_k] \equiv False)) \Rightarrow ({}^qC_j \sim {}^pC_i)$$

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

3. *Specialized definitions* (*subconcept* 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

---

1. We merge ontologies based on similarity relations discussed in the next section. Note that we do not use the term *integrating ontologies* to avoid the wrong impression that any of the communities should agree with or commit to the result of the merging process. Thus, we do not try to detect or resolve mismatches between ontologies as in [20]
2. Intensional definitions are definition of terms by logical axioms. These logical axioms are estimating every *intensional relation* (defined in [9]). For instance, "Faculty" is a conceptual relation and its intensional definition is: ι[Faculty(x)] = Person(x) ∧ (∃y: Course(y) ∧ teaches(x,y))
3. The case of homonyms is not discussed here due to the practical fact that all the terms and their definitions are valid only within the community they are agreed upon. A term outside its community should be uniquely identified according to its respective community [13, 19] - e.g., "education" in Figure 6.

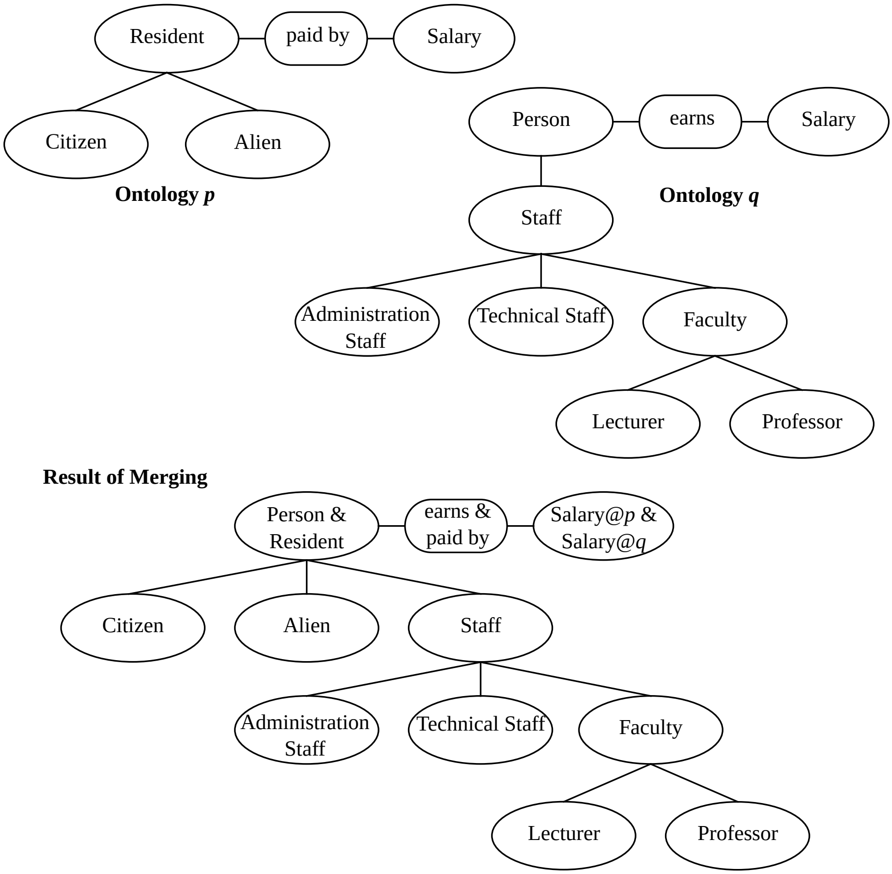

Figure 2. Result of Merging parts of ontologies by finding equal concepts

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.

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

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

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.

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

$$({}^{p}C_i = {}^{q}C_j) \Rightarrow (\forall x)(x \in \varepsilon[{}^{p}C_i] \Leftrightarrow x \in \varepsilon[{}^{q}C_j])$$

The introduced similarity relations can be driven by the 4-intersection approach presented in [4]. One can use logical *and* instead of intersection; *true* and *false* instead of *empty* and *non-*

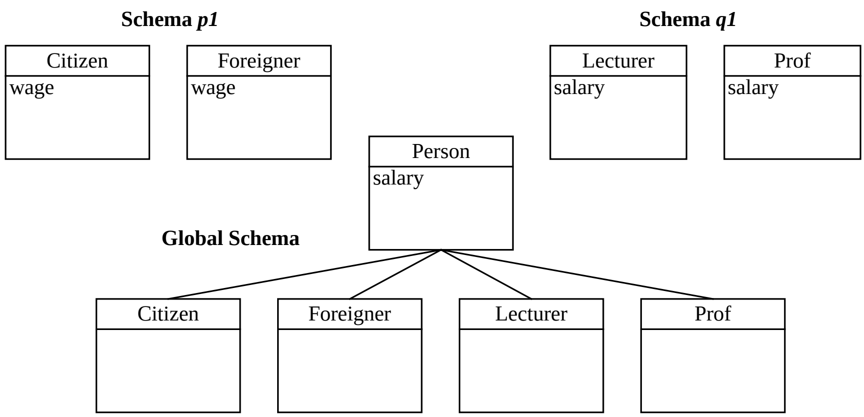

Figure 3. Creating a class Person in global schema.

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 concepts.

Deriving similarities between ontologies requires common references in two ontologies and a reasoning system for matching. The common references can be provided by a higher level ontology such as the library of ontologies on the Ontolingua [6] site 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.

The similarity relations are used to merge two ontologies. We take all the intensional definitions in the respective communities for merging process and explicitly establish similarity relations as explained in the following. Disjoint definitions are not discussed here, and nothing will be done about them in this phase.

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 2[4] (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 4). If C_i specializes any subconcept or subrelation of C_j then 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 4).
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. Although the conjunction of the two intensional definitions may not be proven false, we may not have any instance of such a concept, practically. Yet, if instances of such overlapping concepts exists, the relevance of the new overlapping concept needs supervision of an expert (e.g., concepts “Staff” and “Student” or “Lecturer” and “Graduate student” in Figure 4). Deciding whether such a concept is

4. For sake of simplicity in the next figures, we illustrate only parts of the ontology definitions and parts of the schemas. Dashed lines show the similarity relations established during merging ontologies.

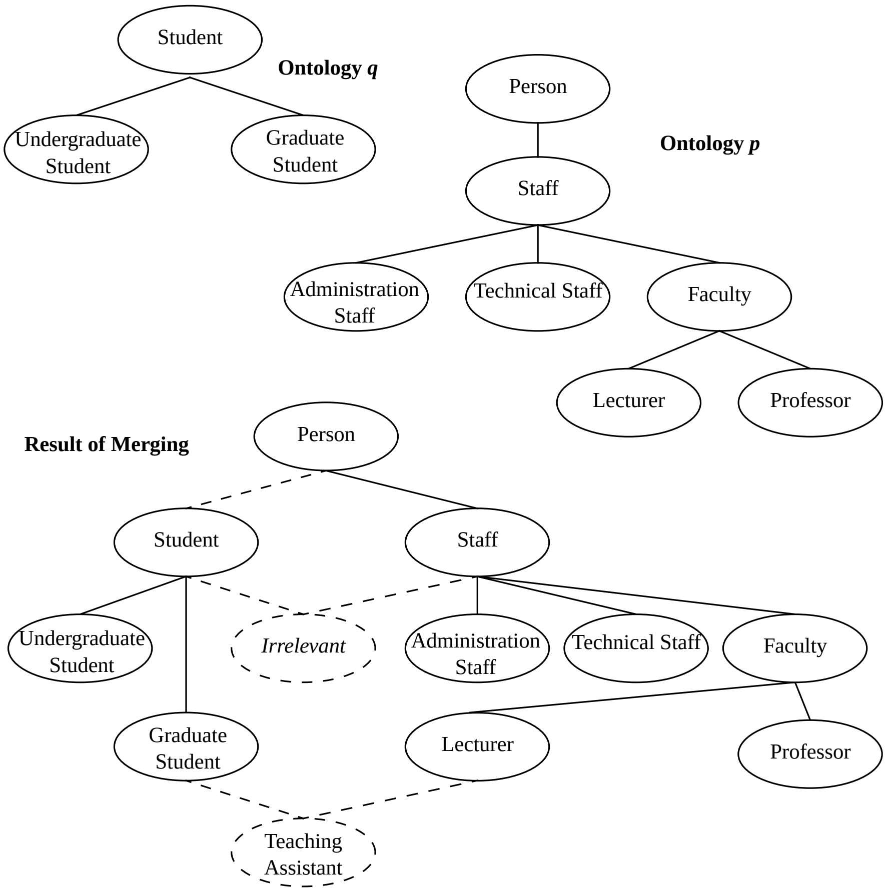

Figure 4. Establishing similarity relations, creating new concept.

relevant for the application domain or not should be done by an expert. Assigning a term to the new concept or relation can be done automatically or by an expert, as well.

5. Schema Integration

In this section, we investigate how the similarity relations are used for the definition of global schemas.

5.1 Derivation of global classes

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 [12]. 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.

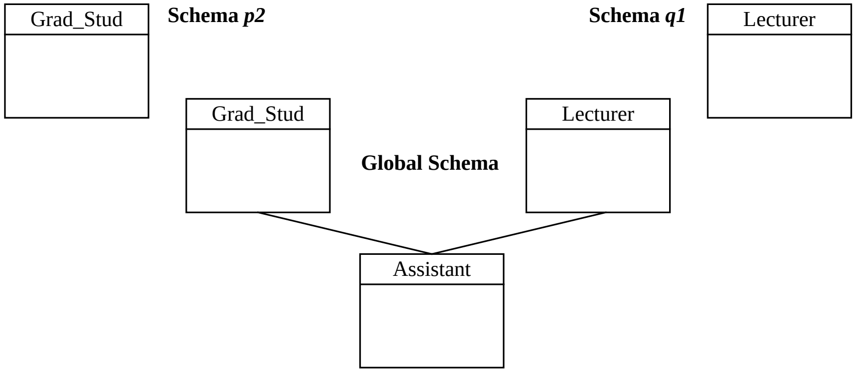

Figure 5. Creating a new class in global schema based on the ontologies in Figure 4.

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 3). The specialized class will be defined by the union of the definition of both classes. 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[5].

If two classes refer to two overlapping concepts, then an overlapping class is added to the global schema based on conjunction concept. 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 5).

Finally, two classes can refer to two overlapping or disjoint concepts, while the corresponding concepts have a common superconcept, such as “Lecturer” and “Citizen” in Figure 3. A class based on the common superconcept is defined in the global schema, e.g., class “Person” in Figure 3.

5.2 Derivation of global class attributes

In the first case, attributes in two classes are referring to the same relation (e.g., relation “earns & paid by” in Figure 2). One attribute will appear in the global class definition representing both attributes (e.g., attribute “salary” and “wage” Figure 3). 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.

Second, two attributes might refer to two relations in a specialization similarity - e.g. education in Figure 6 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

---

5. As an example consider the concept “Lecturer” defined as following:
ι[Lecturer(x)] = Staff(x) $\wedge$ ($\exists$y: Course(y) $\wedge$ teaches(x,y)) $\wedge$ ($\neg$ Professor(x)).
A reasoning system can check an instance of the concept “Person” against this definition and determine whether it may be classified under concept “Lecturer”.

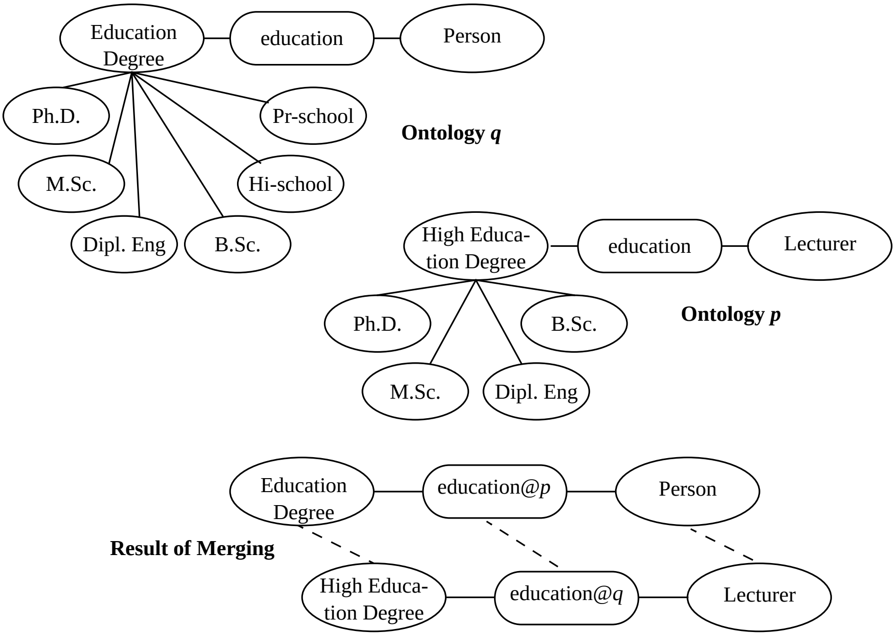

Figure 6. Example of detecting a subrelation similarity in two ontologies.

`narrow`, or “edu-lvl” and “edu-degree” in Figure 7. Classifying the subconcepts may vary in different communities in terms of covered ranges of values or the granularity of classification. In Figure 6 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.

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. Conclusion and Further Work

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

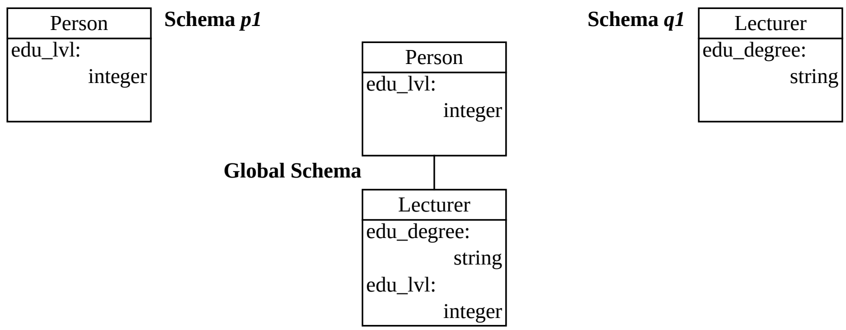

Figure 7. Example of integration of attributes in the global schema level, based on ontologies shown in Figure 6.

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

- We need a clear definition of commitment of a schema to an ontology. By that we are referring to a method to validate commitment of a schema to its ontology. So far, we are using hard links 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.
- 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.

7. References

[1] V. Richard Benjamins and Dieter Fensel. The ontological engineering initiative (KA)^2. In Nicola Guarino, editor, *Formal Ontology in Information Systems*, pages 287–301. IOS Press, 1998.

[2] S. Bergamaschi, S. Castano, S. De Capitani di Vimercati, S. Montanari, and M. Vincini. An intelligent approach to information integration. In Nicola Guarino, editor, *Formal Ontology in Information Systems*, pages 253–267. IOS Press, 1998.

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

[4] Max J. Egenhofer and John R. Herring. Categorizing binary topological relations between regions, lines and points in geographic databases. Technical report, Department of Surveying Engineering, University of Maine, Orono, 1991.

[5] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems*. Addison-Wesley, third edition, 2000.

[6] Adam Farquhar, Richard Fikes, and James Rice. The ontolingua server: a tool for collaborative ontology construction. *International Journal of Human-Computer Studies*, 46:707–

727, 1997. ftp://ftp.ksl.stanford.edu/pub/KSL_Report/KSL_96_26.ps.

[7] Manuel Garcia-Solaco, Felix Saltor, and Malu Castellanos. Semantic heterogeneity in multidatabase systems. In Omran A. Bukhres and Ahmed K. Elmagarmid, editors, *Object-oriented Multidatabase Systems: A Solution for Advanced Applications*, chapter 5, pages 129–202. Printice-Hall, 1996.

[8] Cheng Hian Goh, Stephane Bressan, Stuart Madnick, and Michael Siegel. Context interchange: New features and formalisms for the intelligent integration of information. *ACM Transaction on Information Systems*, 17(3):270–290, 1999.

[9] Nicola Guarino. Formal ontology and information systems. In Nicola Guarino, editor, *Formal Ontology in Information Systems, Proceedings of FOIS'98*, pages 3–17, Trento, Italy, June 1998. IOS Press, Amsterdam.

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

[11] Nicola Guarino and Christopher Welty. Identity, unity, and individuality: Towards a formal toolkit for ontological analysis. In W. Horn, editor, *Proceedings of ECAI-2000 The European Conference on Artificial Intelligence*, Amsterdam, August 2000. IOS Press.

[12] Nicola Guarino and Christopher Welty. Ontological analysis of taxonomic relationships. In Alberto H. F. Laender, Stephan W. Liddle, and Veda C. Storey, editors, *Conceptual Modeling - ER 2000, 19th International Conference Proceedings, Lecture Notes in Computer Science 1920*, pages 210–224. Springr Verlag, October 2000.

[13] Jeff Heflin and James Hendler. Semantic interoperability on the web. In *Extreme Markup Languages 2000*, 2000. http://www.cs.umd.edu/projects/plus/SHOE/pubs/extreme2000.pdf.

[14] Dean Jones. Developing shared ontologies in multi-agent systems. In *ECAI'98 Workshop on Intelligent Information Integration*, Brighton, U.K., August 1998.

[15] Won Kim, Injun Choi, Sunit Gala, and Mark Scheevel. On resolving schematic heterogeneity in multidatabase systems. *Distributed and Parallel Databases*, 1(3):251–277, July 1993.

[16] Won Kim and Jungyun Seo. Classifying schematic and data heterogeneity in multidatabase systems. *IEEE Computer*, 24(12):12–18, December 1991.

[17] Cliff Kottman. *Semantics and Information Communities, The OpenGIS Abstract Specification Topic 14*. OpenGIS Consortium, 35 Main Street, Suite5, Wayland, MA 01778, version 4 edition, April 1999.

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

[19] Pepijn R. S. Visser and Zhan Cui. Heterogeneous ontology structures for distributed architectures. In *ECAI-98 Workshop on Applications of Ontologies and Problem-solving Methods*, pages 112–119, 1998.

[20] Pepijn R. S. Visser, Dean M. Jones, T. J. M. Bench-Capon, and M. J. R. Shave. Assessing heterogeneity by classifying ontology mismatches. In Nicola Guarino, editor, *Formal Ontology in Information Systems, Proceedings of FOIS'98*, pages 148–162, Trento, Italy, 1998. IOS Press.

[21] P.R.S. Visser, D.M. Jones, M.D. Beer, T.J.M. Bench-Capon, B.M. Diaz, and M.J.R. Shave. 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, August 1999.