Merging Expressive Ontologies Using Formal Concept Analysis

Olivier Curé

Université Paris-Est, IGM Terre Digitale, Marne-la-Vallée, France
olivier.cure@univ-paris-est.fr

Abstract. In this paper, we propose a solution to the problem of merging ontologies when instances associated to two source ontologies are available. The solution we propose is based on Formal Concept Analysis (FCA) and considers that ontologies are formalized in expressive Description Logics. Our approach creates a merged ontology which captures the knowledge of the two source ontologies. Contributions of this work are (i) enabling the creation of concepts not originally in the source ontologies, (ii) providing a definition to these concepts in terms of elements of both ontologies and (iii) optimizing the merged ontology. We have studied our approach in the context of spatial information, a domain which exploits many existing ontologies represented with Description Logics.

1 Introduction

The information stored in current IT applications usually need to be exchanged and integrated. These tasks raise several important problems due to format heterogeneity and information uncertainty generally encountered in these applications. Henceforth, we concentrate on Geographical Information Systems (GIS) because they usually integrate ontologies, i.e. a possibly formal representation of a domain of interest, to structure their information. In this paper, we are interested in declarative and logic-based formalisms to represent ontologies. In fact, we consider one of the currently most popular formalism, i.e. Description Logics (DLs). Apart from being popular, and thus offering many open source ontologies on the Web, this representation formalism enables computarized reasoners to infer, usually with sound and complete methods, implicit knowledge from the explicitly represented one.

With so many ontologies being produced, it is inevitable that some of their content overlap and possibly disagree on some concepts. In order to support ontology interoperability, it is required that these ontologies can be semantically related. Thus ontology mediation [7] becomes a main concern. Ontology mediation enables to share data between heterogeneous knowledge bases, and allows applications to reuse data from different knowledge bases. Ontology mediation takes two distinguished forms: (i) Ontology mapping, where the correspondences between elements of two ontologies are stored separately from the ontologies. The correspondences are generally represented using axioms formulated in a peculiar

R. Meersman, P. Herrero, and T. Dillon (Eds.): OTM 2009 Workshops, LNCS 5872, pp. 49–58, 2009.

mapping language. (ii) Ontology merging, which consists in creating a new ontology from the union of source ontologies. The merged ontology is supposed to capture all the knowledge of the sources.

Ontology mediation is an active research field where many kinds of solutions have been proposed: schema-based, instance-based, machine learning-inspired, hybrid approaches; see [9] for a survey on this domain. In this paper, we propose a solution to the ontology merging problem which is based on the techniques of Formal Concept Analysis (FCA) [8]. It extends [3] by dealing with expressive ontologies and their concept descriptions. FCA algorithms are machine learning techniques that enable the creation of a common structure, which may reveal some associations between elements of the two original structures. Thus it requires that some elements from both ontologies can be attached to a same observable item. Starting from this assumption, the processing of our FCA-based algorithms provides a merged ontology.

Our solution extends existing FCA-based systems for ontology merging in the following way: (i) we provide a method to create concepts not originally in the source ontologies, (ii) we define emerging concepts in terms of elements of both ontologies and (iii) we optimize the resulting ontology by eliminating redundant concepts. The step (i) is the classical approach named *ontology alignment* in FCA literature. The steps (ii) and (iii) are an extension of this alignment and exploit concept descriptions and DL reasoner functionalities.

The paper is organized as follows: in Section 2, we present some basic notions about FCA, DLs and present the $\mathcal{ALC}$ description language. In Section 3, we detail our method which enables to create an expressive merged ontology. The main steps are: concept generation, axiomatization of emerging concepts and optimization of the resulting ontology. Section 4 relates our work with existing systems in ontology merging and collaborations between FCA methods and DLs. Section 5 concludes this paper.

2 Basic Notions

FCA is the process of abstracting conceptual descriptions from a set of objects described by attributes [8]. We use some of the methods associated to FCA to merge geographical ontologies. Intuitively, this means that we merge two ontologies in a context consisting of a set of objects, a set of attributes, one for each ontology, and a set of correspondences between objects and attributes. FCA is based on the notion of a *formal context*.

Definition 1. A formal context is a triple $\mathcal{K} = (G, M, I)$, where G is a set of objects, M is a set of attributes and I is a binary relation between G and M, i.e. $I \subseteq G \times M$. For an object g and an attribute m, $(g, m) \in I$ is read as "object g has attribute m".

Given a formal context, we can define the notion of formal concepts:

Definition 2. For $A \subseteq G$, we define $A' = \{m \in M | \forall g \in A : (g, m) \in I\}$ and for $B \subseteq M$, we define $B' = \{g \in G | \forall m \in B : (g, m) \in I\}$. A formal concept of $\mathcal{K}$ is defined as a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$.

The hierarchy of formal concepts is formalized by $(A_1, B_1) \leq (A_2, B_2) \iff A_1 \subseteq A_2$ and $B_1 \subseteq B_2$. The concept lattice of $\mathcal{K}$ is the set of all its formal concepts with the partial order $\leq$.

DLs are a family of knowledge representation formalisms allowing to reason over domain knowledge, in a formal and well-understood way. Central DL notions are concepts (unary predicates), roles (binary predicates) and individuals. A concept represents a set of individuals while a role determines a binary relationship between concepts. DLs are a fragment of first-order logic and thus concepts and roles are designed according to a syntax and a semantics. Some of the main assets of this family of formalims are decidability, efficient reasoning algorithms and the ability to propose a hierarchy of languages with various expressive power.

A key notion in DLs is the separation of the terminological (or intensional) knowledge, called a TBox, to the assertional (or extensional) knowledge, called the ABox. The TBox is generally considered to be the ontology. Together, a TBox and a ABox represent a Knowledge Base (KB), denoted $KB = \langle TBox, ABox \rangle$.

The TBox is composed of "primitive concepts" which are ground descriptions that are used to form more complex descriptions, "defined concepts" which are designed using a set of constructors of the description language, e.g. conjunction($\sqcap$), disjunction ($\sqcup$), negation ($\neg$), universal ($\forall$) and existential ($\exists$) value quantifiers, etc.

The description language we are using in this paper correspond to $\mathcal{ALC}$ (Attributive Language with Complements). Concept descriptions in this language are formed according to the following syntax rule, where the letter A is used for atomic concepts, the letter R for atomic roles and the letters C and D for concept descriptions:

$C, D ::= \bot \mid \top \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C$

The semantics generally adopted for the $\mathcal{ALC}$ language is based on Tarski-style semantics and we invite the interested reader to study [1] for details.

In DLs, the basic reasoning service on concept expressions is *subsumption*, written $C \sqsubseteq D$. This inference checks whether the first concept always denotes a subset of the set denoted by the second one. We use this service on the optimization of merged ontologies.

Both domains, FCA and DL ontologies, use the notion of *concept*. In the rest of this paper, concepts in the context of FCA (resp. DL ontology) are named formal concepts, resp. DL concepts. To clarify the distinction between them, we can state that DL concepts correspond to the attributes of $\mathcal{K}$.

3 Ontology Merging Using FCA

Let consider 2 geographical applications that manipulate space parcel data. Each application uses an independent ontology formalism to represent the concepts related to its data. Also the teams of experts that designed each ontology may not agree on the semantics of some concepts. Anyhow, the 2 applications need to exchange information, and thus require that some correspondences are discovered

between their DL concepts. The following 2 ontology extracts, O_1 and O_2, are used all along this paper. In order to ease the understanding and reading of our example, all concepts and roles are underscripted with the number of their respective ontology, i.e. 1 for O_1 and 2 for O_2.

Terminological axioms of ontology O_1

1. $CF_1 \equiv F_1 \sqcap \exists vegetation_1.C_1$
2. $BLF_1 \equiv F_1 \sqcap \exists vegetation_1.M_1$
3. $C_1 \sqcap M_1 \sqsubseteq \bot$

This extract of ontology O_1 defines 2 concepts, CF_1, standing for Coniferous Forest, and BLF_1, standing for Broad Leaved Forest, in terms of the concepts F_1 (Forest), C_1 (Coniferophyta) and M_1 (Magnoliophyta). Line #1 states that the coniferous forest concept is defined as the intersection of the concept Forest of O_1 and the concept having at least one vegetation being a coniferophyta. Line #2 defines the concept of a broad leaved forest accordingly with magnoliophyta. Line #3 states that the concepts coniferophyta and magnoliophyta are disjoint.

Terminological axioms of ontology O_2

4. $CF_2 \equiv F_2 \sqcap \forall vegetation_2.C_2 \sqcap \ \exists vegetation_2.C_2$
5. $BLF_2 \equiv F_2 \sqcap \forall vegetation_2.M_2 \sqcap \ \exists vegetation_2.M_2$
6. $MF_2 \equiv F_2 \sqcap \exists vegetation_2.C_2 \sqcap \ \exists vegetation_2.M_2$
7. $C_2 \sqcap M_2 \sqsubseteq \bot$

The study of O_2 emphasizes that designers do not entirely agree on the semantics of forest related concepts of O_1. On line #4, the concept of a coniferous forest is defined has being a forest composed of at least coniferophyta vegetation and exclusively of this kind of vegetation. Line #5 defines the concept of broad leaved forest accordingly with magnoliophyta. In order to represent other kinds of forests, the designers of O_2 define a mixed forest concept as the intersection of being a forest with at least one coniferophyta vegetation and at least one magnoliophyta vegetation. Finally Line #8 states that the concepts coniferophyta and magnoliophyta of O_2 are disjoint.

We consider DL knowledge bases with non-empty TBoxes and ABoxes. In a first step, we map the information of the 2 ABoxes on a common set of observed objects. The information of these ABoxes can be stored in a structured or unstructured format. It is interesting to note the activity of several research teams in the DL and Semantic Web community in studying cooperations between the domains of databases and knowledge bases represented in a DL. For instance, the authors of [13] recently claimed that the ideal solution would be to have the individuals of the ABox stored in a relational databases and represent the schema of this database in a DL TBox. Also tackling this same objective, the team supporting the Pellet reasoner, one of the most popular OWL reasoner, recently released *OWLgres* which is being defined by their creators as a 'scalable reasoner for OWL2'. A main objective of this tool is to provide a conjunctive query answering service using SPARQL and the performance properties

of relational database management systems. Using such an approach, the set of observed objects may be retrieved from existing relational database instances using already existing tools of FCA adapted to relational databases.

The mapping we propose between both ontologies can be represented by a *matrix*, either generated by a specific tool and/or by interactions with end-users. In order to map concepts of both ontologies via the selected set of observed objects, a reference reconciliation tool may be used [5].

We present a sample of this mapping in Table 1: the rows correspond to the objects of $\mathcal{K}$, i.e. common instances of the KB's ABox, and are identified by integer values from 1 to 6 in our example. The columns correspond to FCA attributes of $\mathcal{K}$, i.e. concept names of the 2 TBoxes. In the same table, we present, side by side, the formal concepts coming from our 2 ontologies, i.e. CF_1, BLF_1, F_1 from O_1, and CF_2, BLF_2, MF_2, F_2 from O_2.

3.1 Generation of a Merged Lattice

The matrix is built using the information stored in the TBox and ABox of both ontologies:

– first, for each row, mark the columns where a specific instance is observed, e.g. the object on line 1 is an instance of the CF_1 and CF_2 concepts. Thus ABox information is used in this step.
– then, complete the row with the transitive closure of the subsumption relation between ontology concepts, e.g.: line 1 must be also marked for DL concepts F_1 and F_2, as respective ontologies state that: $CF_1 \sqsubseteq F_1$ and $CF_2 \sqsubseteq F_2$. Here, the concept hierarchy of TBoxes are exploited.

It is interesting to note the lines #3 and #6 emphasize different assumption for their respective parcels. For instance, the parcel conrresponding to line #3 has been defined as a coniferous forest using the classification of O_1 while, possibly due to a vegetation not limited to coniferophyta, it has been defined as a mixed forest using O_2. The same kind of approach applies to the parcel associated to line #6.

Using Table 1 with the Galois connection method [4], we obtain the lattice of Figure 1, where a node contains two sets: a set of objects (identified by the integer values of the first column of our matrix) from $\mathcal{K}$ (extension), and a set

Table 1. Sample dataset for our ontology merging example

	CF_1	BLF_1	F_1	CF_2	BLF_2	MF_2	F_2
1	x		x	x			x
2	x		x	x			x
3	x		x			x	x
4		x	x		x		x
5		x	x		x		x
6		x	x			x	x

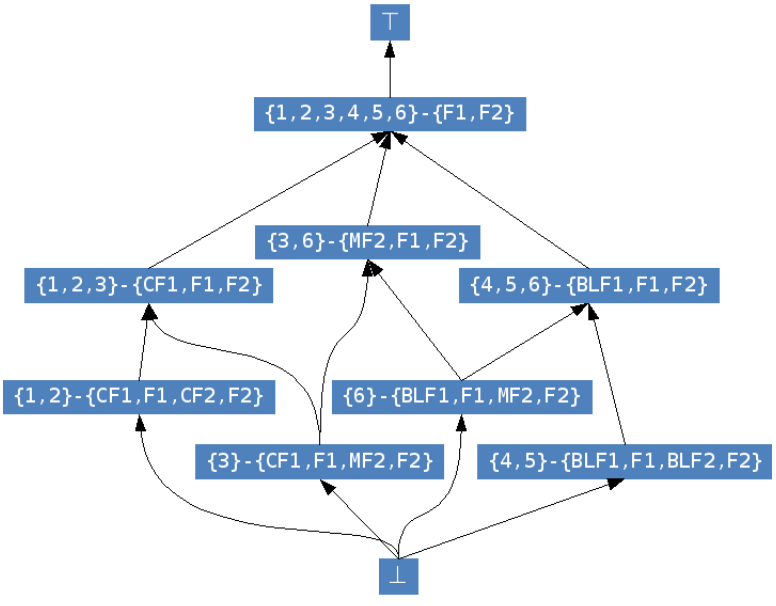

Fig. 1. Galois connection lattice

of DL concepts from the source ontologies (intension), identified by the concept labels of source ontologies.

3.2 Dealing with Emerging Concepts

We now consider that the extensional part is not useful to comprehend a node. Thus, we remove it from all the nodes of the lattice, and only concept names remain (the intensional part). Then, we can also remove the redundancy, by deleting repeated occurences of a given concept name along a path of the lattice. The approach that we use is based on a bottom-up navigation of the lattice nodes: start from the bottom ($\perp$) and navigate upwards. For each node, analyze the set of concept names, and eliminate names that are present in the set of one of its direct successor, i.e. node above it and reached using a unique edge. This method has been adopted due to the lattice structure obtained by applying the Galois connection method. Finally we obtain Figure 2 where lattice nodes contain a single set, corresponding to concept names from one of the two original ontologies. We now classify the kinds of node sets we encounter:

1. a singleton: a name of a concept from either original ontology, because it can be distinguished from any of its successors by this specific name, e.g.: $\{CF_1\}$.
2. an empty node, because it can't be directly distinguished from any of its possible successors. We identify each of these nodes with a unique symbol disjoint from the concept symbols of the source ontologies. In Figure 2, we have 2 such nodes which are named α and β.

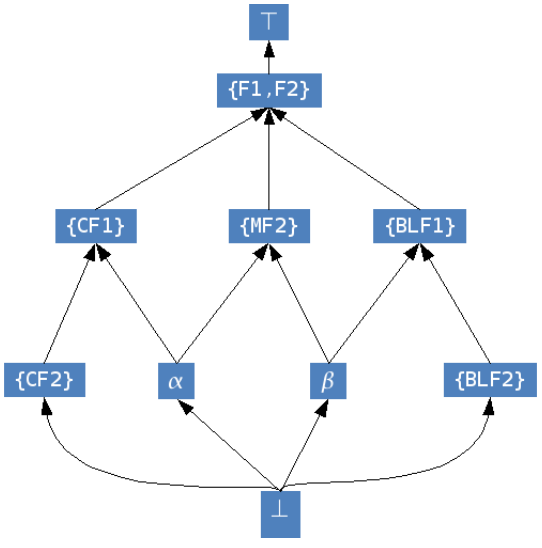

Fig. 2. Galois connection lattice with "empty nodes"

3. a set of several names, all belonging to source ontologies, because the mediation based on the given two ABoxes, has not been able to split names. Indeed, it is as if the names are glued together in a single concept name.

All singletons are maintained in the resulting merged ontology and we are now aiming to provide a concept description to the remaining concepts, case 2 and 3 of our node classification. The first step toward our solution is to expand the concepts of the merged ontology according to their respective TBoxes. That is, we replace each occurence of a name on the right hand-side of a definition by the concepts that it stands for. A prerequisite of this approach is that we are dealing with acyclic TBoxes. Thus this process stops and the resulting descriptions contain only primitive concepts on the right hand-side.

We first deal with the nodes which are formed of several concept symbols, denoted σ_i, e.g. node labelled F_1, F_2 in Figure 2. Due to the algorithm adopted from the generation of the Galois connection lattice [4], these nodes appear at the top of the lattice and do not have multiple inheritance to concepts that are not of this form. Thus we adopt a top-down approach from the top concept of our merged ontology. We consider that the concepts associated are equivalent, e.g. $F_1 \equiv F_2$, propose a single concept symbol σ, e.g. F (Forest) for F_1, F_2, and associate information to this concept stating that this concept is equivalent to the original concepts for interoperability reasons, e.g. $F \approx F_1$ and $F \approx F_2$. Now all occurences of the concept σ_i are translated into the concept symbol σ in the concept descriptions of the merged ontology.

We can now concentrate on empty nodes, e.g. α and β. Again, according to the Galois based lattice creation, these nodes can not be at the root of the lattice. This means that they inherit from some other concept(s). We use the description

of these inherited concept(s) to provide a description. Using this method, the concepts α and β of Figure 2 have the following descriptions:

$\alpha \equiv CF_1 \sqcap MF_2 \equiv \exists vegetation_1.C_1 \sqcap \exists vegetation_2.C_2 \sqcap vegetation_2.M_2$
$\beta \equiv BLF_1 \sqcap MF_2 \equiv \exists vegetation_1.M_1 \sqcap \exists vegetation_2.C_2 \sqcap vegetation_2.M_2$

All concepts from the merged ontology have been associated to a concept description, except of course the primitive concepts. But we can do more and optimize the descriptions. This optimization operation is supported by the possible alignement we can perfom on the primitive concepts of both ontologies O_1 and O_2. We mainly distinguish between 3 situations:

- a situation where primitive concepts are coming from a single integrated ontology. In our example, this means that we can state that $C_1 \equiv C_2$, $M_1 \equiv M_2$ and even $vegetation_1 \equiv vegetation_2$.
- a situation where primitive concepts of source ontologies are not aligned natively. Then, we can use an external tool and/or interactions with end-users to align them. In our example, we would end up with the same equivalence relations as in the previous case.
- no alignment on primitive concepts is possible and our merged ontology can not be optimized.

In our example, for the first 2 situations the descriptions of the concepts of our merged ontology are (assuming a renaming of the primitive concepts, e.g. $C \equiv C_1 \equiv C_2$ and $M \equiv M_1 \equiv M_2$):

8. $CF_1 \equiv F \sqcap \exists vegetation.C$
9. $BLF_1 \equiv F \sqcap \exists vegetation.M$

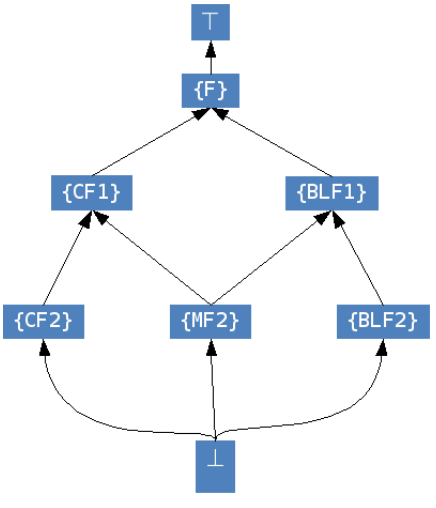

Fig. 3. Galois connection lattice with generated labels

10. $CF_2 \equiv CF_1 \sqcap \forall vegetation.C \sqcap \exists vegetation.C$
11. $BLF_2 \equiv BLF_2 \sqcap \forall vegetation.M \sqcap \exists vegetation.M$
12. $MF_2 \equiv F \sqcap \exists vegetation.C \sqcap \exists vegetation.M$
13. $alpha \equiv F \sqcap \exists vegetation.C \sqcap \exists vegetation.M$
14. $beta \equiv F \sqcap \exists vegetation.C \sqcap \exists vegetation.M$
15. $C \sqcap M \sqsubseteq \bot$

Our merged ontology can now be classified using a DL reasoner, e.g. Fact, Racer. This processing enables to find some new subsumption relations which enable to design the ontology of Figure 3.

Figure 3 shows the merged ontology resulting from application of FCA.

4 Related Work

In this Section, we survey related works in ontology mediation solutions and in particular we present some solutions which exploit extensions of the ontologies, i.e. ABoxes.

In the literature, two distinct approaches in ontology merging have been distinguished. In the first approach, the merged ontology captures all the knowledge of the source ontologies and replaces them. An example of such a system is presented in [12] with the PROMPT tool. In the second approach the source ontologies are not replaced by the merged, but rather a so-called 'bridge ontology' is created. The bridge ontology imports the original ontologies and defines the correspondences using axioms which are called "bridge axioms". An example of such an approach is the Ontomerge solution which has been described in [6].

The most relevant work related to our solution is the FCA-merge system [14]. It uses instances of ontology classes to exploit an FCA algorithm. The FCA-merge system produces a lattice of concepts which relates concepts from the source ontologies. This new concept lattice is then handed to the domain expert in order to generate the merged ontology. Thus we can consider FCA-merge to be a semi-automatic solution while our solution aims to generate the merged ontology automatically. So the main differences are that the FCA-merge is unable to propose concepts emerging from the fusion of the source ontologies and does not propose a label generation solution. Also, without the help of domain experts, the FCA-merge system is not able to refine the merged ontology.

Considering works involving FCA methods and DLs, it is interesting to study [2]. In this paper the authors are concerned with the completeness quality dimension of TBoxes, i.e. they propose techniques to enable ontology engineers in checking if all the relevant concepts of an application domain are present in a TBox. Like our approach, one of their concern is to minimize interactions with domain experts. Hence FCA techniques are being used to withdraw trivial questions that may be asked to experts in case of incomplete TBoxes. The approach we presented in this paper is more concern with the generation and optimization of mediated ontology. And we can consider that our approach is more involved in the soundness quality dimension and tackles the issue of generating different forms of merged ontology.

5 Conclusion

In this paper, we presented an approach to merge ontologies based on the methods of FCA. Our main contribution enables (i) the creation of concepts not originally in the source ontologies, (ii) the definition of the concepts in terms of elements of both ontologies and (iii) the optimization of the resulting ontology by eliminating redundant concepts.

Future work on this system are related to extracting automatically a valuable and minimal set of instances from ABoxes for the Galois connection matrix and studying expressive DL beyond the $\mathcal{ALC}$ language.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, New York (2003)
2. Baader, F., Ganter, B., Sertkaya, B., Sattler, U.: Completing Description Logic Knowledge Bases Using Formal Concept Analysis. In: Proc. IJCAI 2007, pp. 230–235 (2007)
3. Curé, O., Jeansoulin, R.: An FCA-based Solution for Ontology Mediation. In: Proc. CIKM workshops (2008) (to appear)
4. Davey, B., Priestley, H.: Introduction to lattices and Order. Cambridge University Press, New York (2002)
5. Dong, X., Halevy, A., Madhavan, J.: Reference reconciliation in complex information spaces. In: Proc. SIGMOD 2005, pp. 85–96 (2005)
6. Dou, D., McDermott, D., Qi, P.: Ontology translation by ontology merging and automated reasoning. In: Proc. EKAW 2002, pp. 3–18 (2002)
7. Ehrig, M.: Ontology Alignment: Bridging the Semantic Gap. Springer, New York (2006)
8. Ganter, B., Wille, R.: Formal Concept Analysis: mathematical foundations. Springer, New York (1999)
9. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. Knowledge Engineering Review 18(1), 1–31 (2003)
10. Kanellakis, P.C.: Elements of relational database theory. In: Handbook of theoretical computer science. Formal models and semantics, vol. B, pp. 1073–1156. MIT Press, Cambridge (1990)
11. Motik, B., Horrocks, I., Sattler, U.: Bridging the gap between OWL and relational databases. In: Proc. WWW 2007 (2007)
12. Noy, N., Musen, M.: PROMPT: Algorithm and tool for automated ontology merging and alignment. In: Proc. AAAI 2000 (2000)
13. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking Data to Ontologies. Journal of Data Semantics 10, 133–173 (2008)
14. Stumme, G., Maedche, A.: FCA-MERGE: Bottom-Up Merging of Ontologies. In: Proc. IJCAI 2001, pp. 225–234 (2001)