



**HAL**  
open science

# Combining Concept Annotation and Pattern Structures for Guiding Ontology Mapping

Pierre Monnin, Amedeo Napoli, Adrien Coulet

► **To cite this version:**

Pierre Monnin, Amedeo Napoli, Adrien Coulet. Combining Concept Annotation and Pattern Structures for Guiding Ontology Mapping. FCA4AI@IJCAI2018 - 6th International Workshop "What can FCA do for Artificial Intelligence?", Jul 2018, Stockholm, Sweden. hal-01858391

**HAL Id: hal-01858391**

**<https://inria.hal.science/hal-01858391v1>**

Submitted on 20 Aug 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Combining Concept Annotation and Pattern Structures for Guiding Ontology Mapping

Pierre Monnin<sup>1</sup>, Amedeo Napoli<sup>1</sup>, and Adrien Coulet<sup>1,2</sup>

<sup>1</sup> LORIA (CNRS, Inria Nancy-Grand Est, Université de Lorraine),  
Vandœuvre-Lès-Nancy, France

`pierre.monnin@loria.fr`

<sup>2</sup> BMIR, Stanford University, USA

**Abstract.** Formal Concept Analysis (FCA) is a mathematical framework classifying in formal concepts a set of objects w.r.t. their common attributes. To this aim, FCA relies on a binary incidence relationship indicating whether an object has an attribute. On one hand, in order to consider more complex descriptions for objects (*e.g.*, intervals, graphs), FCA has been extended with Pattern Structures. On the other hand, in a previous work, we introduced the notion of Concept Annotation, adding a third dimension to formal concepts, computed over the extent, without modifying the original classification. In this paper, we combine Concept Annotation with the formalism of Pattern Structures and we consider multiple annotation possibilities, *i.e.*, multiple annotations for one concept and computing the annotation over the intent. We illustrate our approach and its interest with two use cases: *(i)* suggesting mappings between ontology classes and *(ii)* finding specific classes frequently associated as domain and range of a predicate.

**Keywords:** Concept Annotation · Pattern Structures · Formal Concept Analysis

## 1 Introduction

An ontology is a formal representation of a particular domain [8], consisting of two parts: an *assertion component* (*ABox*) and a *terminological component* (*TBox*). In the *TBox*, classes and predicates between classes are defined while in the *ABox*, individuals instantiate classes and predicates. For example, the drug *codein* can be considered as an individual instantiating the class *Analgesics*.

Nevertheless, there is a need for structure between ontologies. Indeed, individuals of an *ABox* may instantiate classes from several ontologies. For example, in the medical domain, individuals representing diseases may instantiate classes of several ontologies of diseases, such as MeSH (*Medical Subject Headings*), ICD-9-CM and ICD-10-CM (*International Classification of Diseases version 9 and 10, Clinical Modification*). Corresponding classes from distinct ontologies may be mapped thanks to equivalence relationships resulting from an alignment process [5]. This alignment may be either a manual review by an expert or a semi-automatic process.

On the other hand, within an ontology, additional knowledge and structuring may be discovered. Thus, the *ABox* of an ontology may also contain predicate assertions, *i.e.*, relations between individuals. For example, in DrugBank, a database of drug information, a drug can be associated with a gene, indicating an interaction between them. Thereby, the gene *VKORC1* is indicated to be an inhibitor of the drug *warfarin*. Genes and drugs instantiate classes of ontologies, such as ATC (*Anatomical Therapeutic Chemical Classification System*) for drugs and GO (*Gene Ontology*) for genes. Therefore, from the predicate assertions, we could discover classes of drugs and genes frequently associated as domain and range of a predicate. Such domain and range associations could be interesting as they may indicate common properties of the gene class or the drug class.

In this article, we aim at addressing both of the aforementioned use cases, *(i)* suggesting equivalence relationships between ontology classes and *(ii)* discovering classes frequently associated in domain and range of a predicate, with Formal Concept Analysis (FCA) [7]. FCA is a well-founded mathematical framework adapted to knowledge engineering purposes [1, 3, 4] that groups objects w.r.t. their common attributes in formal concepts. The association of an attribute to an object is expressed thanks to a binary relationship. Formal concepts are organized by a partial order in a hierarchical structure called a lattice. In a previous work [9], we introduced an extension of FCA, called *Concept Annotation*, allowing to compare the hierarchical structure of the lattice with the class hierarchy of an ontology for possible refactoring. The main interest of Concept Annotation resides in adding a third dimension to a previously generated lattice without changing its structure. The main contribution of the present article is to extend the initial definition of Concept Annotation by *(i)* associating multiple Concept Annotations with formal concepts, and *(ii)* using a formalism similar to Pattern Structures [6] to express the annotations. Pattern Structures are another extension of FCA dealing with objects having complex (non binary) descriptions (*e.g.*, graphs, numerical values, classes of an ontology). In the following, we assume that the reader is familiar with the basics of FCA and Pattern Structures.

This article is organized as follows. In Section 2, necessary basics about ontologies are recalled. In Section 3, we present the initial definition of Concept Annotation as well as our proposed extension. Then, in Sections 4 and 5, we illustrate the combination of Concept Annotation and Pattern Structures by, respectively, suggesting equivalence relationships between classes and discovering classes frequently associated as domain and range of a predicate. Finally, in Sections 6 and 7, we discuss the results we obtain on the two considered use cases as well as the next challenges to tackle.

## 2 Basics about Ontologies

An ontology is a formal representation of a particular domain [8]. It is composed of two parts, the *TBox* and the *ABox*. The *TBox* defines classes and relationships between them. We denote  $\mathcal{C}(\mathcal{O})$  the set of classes of the ontology  $\mathcal{O}$ . Classes of an ontology are instantiated by individuals of the *ABox*. These individuals can also

be involved in instantiation of relationships (such as the drug-gene relationship in DrugBank previously explained).

Classes of an ontology are organized in a partial order by a subsumption relation denoted by  $\sqsubseteq$ . Considering two classes  $C$  and  $D$ ,  $C \sqsubseteq D$  states that every instance of  $C$  is also an instance of  $D$ . The *least common subsumer* (sometimes named the *lowest common ancestor*) of two classes  $C_1$  and  $C_2$  of an ontology is the most specific class subsuming both  $C_1$  and  $C_2$ . It is denoted by  $\text{lcs}(C_1, C_2)$ . In  $\mathcal{EL}$  ontologies where no cycle appears, the lcs of two classes always exists [2]. Considering a set of classes  $\mathcal{C}_n = \{C_1, C_2, \dots, C_n\}$ , we define the set of the most specific classes of  $\mathcal{C}_n$  as  $\min \mathcal{C}_n = \{C \in \mathcal{C}_n \mid \nexists D \in \mathcal{C}_n, D \sqsubseteq C\}$ .

### 3 Concept Annotation

Concept Annotation is an extension of FCA that we introduced in a previous work [9]. In this section, we first recall basics about our initial work and then present the proposed extension.

#### 3.1 Basics about Concept Annotation

In Concept Annotation, standard FCA is applied on a formal context to build a concept lattice. Then, for each formal concept, an annotation is computed, adding a third dimension to each concept.

For example, in [9], a lattice is built from a formal context  $(G, M, I)$  where  $G$  is a set of individuals from the *ABox* of an ontology  $\mathcal{O}$  and  $M$  is a set of predicates of this ontology.  $(g, m) \in I$  indicates that the individual  $g$  is involved in a relationship whose predicate is  $m$ . Then, classes of the *TBox* instantiated by the individuals in  $G$  are added as annotation. The resulting hierarchy formed by the lattice and the annotations is compared to the class hierarchy of the ontology. To compute the annotation, we defined a derivation operator  $(\cdot)^\diamond : 2^G \rightarrow 2^{\mathcal{C}(\mathcal{O})}$ , where  $2^{\mathcal{C}(\mathcal{O})}$  corresponds to the powerset of the set of classes of the ontology. Considering a formal concept  $(A, B)$  where  $A$  is a set of individuals, the annotation is defined thanks to the derivation operator as follows:  $A^\diamond = \bigcap_{g \in A} \{C \mid \mathcal{O} \models C(g)\}$ <sup>3</sup>. Intuitively, it corresponds to the set of all classes of  $\mathcal{O}$  instantiated by all individuals in  $A$ .

#### 3.2 Combining Concept Annotation and Pattern Structures

In this paper, we propose to combine Concept Annotation with Pattern Structures. Indeed, such combination will allow us to only keep as annotation the most specific classes instantiated by all individuals.

To do so, we define a function  $\delta : 2^G \rightarrow 2^{\mathcal{C}(\mathcal{O})}$  that associates an individual from  $G$  with the set of the most specific classes of the ontology  $\mathcal{O}$  that this individual instantiates. Formally, given  $g \in G$ ,  $\delta(g) = \min \{C \in \mathcal{C}(\mathcal{O}) \mid \mathcal{O} \models C(g)\}$ .

<sup>3</sup>  $C(g)$  indicates that  $g$  is an instance of  $C$  in the Description Logics formalism

$\delta(g)$  is considered as the description of the individual  $g$ . Given two individuals,  $g_1, g_2 \in G$ , we define a similarity operator  $\sqcap$  to compare their two descriptions as follows:

$$\delta(g_1) \sqcap \delta(g_2) = \min \{ \text{lcs}(C_1, C_2) \mid \forall (C_1, C_2) \in \delta(g_1) \times \delta(g_2) \}$$

The use of the *least common subsumer* allows to compute the most specific classes of the ontology that both  $g_1$  and  $g_2$  instantiates. Finally, given a formal concept  $(A, B)$ , we annotate it thanks to the new derivation operator  $(\cdot)^\circ : 2^G \rightarrow 2^{\mathcal{C}(\mathcal{O})}$  defined as  $A^\circ = \prod_{g \in A} \delta(g)$ .

In the following sections, the defined functions and operators will be restricted to specific ontologies. Thus,  $\delta_i$ ,  $\sqcap_i$  and  $(\cdot)^{\circ_i}$  will only be applicable to classes of an ontology  $\mathcal{O}_i$ .

## 4 Suggesting Mappings between Classes of Ontologies

In this section, we illustrate the combination of Concept Annotation and Pattern Structures with the use case of suggesting equivalence relationships between classes of two ontologies, denoted by  $\mathcal{O}_1$  and  $\mathcal{O}_2$ . To generate these mappings, we consider individuals that instantiate classes of both  $\mathcal{O}_1$  and  $\mathcal{O}_2$ . They also instantiate classes of another ontology, denoted by  $\mathcal{O}_{ref}$ , that is considered in this setting as the reference ontology, *i.e.* the feature we consider to build the original classification of the set of instances. In order to avoid complexity problems, we keep  $\mathcal{O}_{ref}$  of a small size.

### 4.1 Classifying Individuals w.r.t. $\mathcal{O}_{ref}$ in a Concept Lattice

The first step is to build a concept lattice classifying the individuals w.r.t. the classes of  $\mathcal{O}_{ref}$  they instantiate. To this aim, we use the pattern structure  $(G, (2^{\mathcal{C}(\mathcal{O}_{ref})}, \sqcap_{ref}), \delta_{ref})$ .  $G$  is the set of individuals and  $2^{\mathcal{C}(\mathcal{O}_{ref})}$  is the powerset of the set of classes of  $\mathcal{O}_{ref}$ .  $\delta_{ref}$  and  $\sqcap_{ref}$  are defined as in Subsection 3.2 w.r.t. the classes of  $\mathcal{O}_{ref}$ . From this pattern structure definition, we obtain pattern concepts  $(A, D)$  organized in a concept lattice, where  $A \subseteq G$  is a set of individuals and  $D \in 2^{\mathcal{C}(\mathcal{O}_{ref})}$  is a set of the most specific classes of  $\mathcal{O}_{ref}$  that all individuals in  $A$  instantiate.

### 4.2 Annotating the Concept Lattice with $\mathcal{O}_1$ and $\mathcal{O}_2$

In this next step, classes from  $\mathcal{O}_1$  and  $\mathcal{O}_2$  are considered for annotating the concept lattice. To this aim, we define *two* annotations (one per ontology) using the formalism defined in Subsection 3.2. Thereby, we use two functions  $\delta_1$  and  $\delta_2$  to associate an individual  $g$  with the set of the most specific classes from  $\mathcal{O}_1$  and  $\mathcal{O}_2$  that this individual instantiates.  $\sqcap_1$  and  $\sqcap_2$  are used to compute the similarity between descriptions of two individuals w.r.t. the two considered ontologies  $\mathcal{O}_1$  and  $\mathcal{O}_2$ . Finally, to compute the two annotations for each pattern concept  $(A, D)$ ,

two derivation operators  $(\cdot)^{\circ_1} : 2^G \rightarrow 2^{\mathcal{C}(\mathcal{O}_1)}$  and  $(\cdot)^{\circ_2} : 2^G \rightarrow 2^{\mathcal{C}(\mathcal{O}_2)}$  are defined as follows:

$$A^{\circ_1} = \prod_{g \in A} \delta_1(g) \quad A^{\circ_2} = \prod_{g \in A} \delta_2(g)$$

As a result, for each pattern concept  $(A, D)$ , an annotated pattern concept  $(A, D, A^{\circ_1}, A^{\circ_2})$  is obtained where  $A^{\circ_1}$  (resp.  $A^{\circ_2}$ ) is the set of the most specific classes of  $\mathcal{O}_1$  (resp.  $\mathcal{O}_2$ ) that all individuals in  $A$  instantiate.

### 4.3 Reading Mappings from the Lattice

Let us consider an annotated pattern concept  $(A, D, A^{\circ_1}, A^{\circ_2})$ . From the previous definitions, we know that  $A^{\circ_1} \subseteq \mathcal{C}(\mathcal{O}_1)$  contains the set of the most specific classes of  $\mathcal{O}_1$  that all individuals in  $A$  instantiate. Similarly,  $A^{\circ_2} \subseteq \mathcal{C}(\mathcal{O}_2)$  contains the set of the most specific classes of  $\mathcal{O}_2$  that all individuals in  $A$  instantiate. Therefore, considering each pair of classes  $(C_1, C_2) \in A^{\circ_1} \times A^{\circ_2}$ , we know that they are instantiated by the same set of individuals, *i.e.*, individuals in  $A$ . Therefore, an equivalence relationship is suggested between  $C_1$  and  $C_2$ , based on the instances of the two classes.

## 5 Discovering Associated Classes as Domain and Range of a Predicate

In this section, we illustrate how Concept Annotation and Pattern Structures could be used to discover the most specific classes frequently associated as domain and range of a predicate from its instantiations in an *ABox*. Such domain and range characterization could indeed indicate common behavior at the class level. For example, considering DrugBank, a gene can be involved with a drug in a relationship whose action is specified (*e.g.*, inhibitor, antagonist). Such relationships correspond to assertions of a predicate. As drugs instantiate classes of ATC and genes instantiate classes of GO, we could discover that instances of a GO class, considered as a family of genes, are frequently indicated as inhibitors of instances of an ATC class, considered as a family of drugs.

In the following paragraphs, we consider individuals, instantiating classes of an ontology  $\mathcal{O}_1$ , that are involved in relationships with other individuals, instantiating classes of an ontology  $\mathcal{O}_2$ <sup>4</sup>.

### 5.1 Classifying Relationships in a Concept Lattice

First, a classification of relationships is established. To this aim, we consider the formal context  $(G, M, I)$  where  $G$  is the set of individuals instantiating classes from  $\mathcal{O}_1$  and  $M$  is the set of individuals instantiating classes from  $\mathcal{O}_2$ . Given  $g \in G$ , and  $m \in M$ ,  $(g, m) \in I$  if and only if a relationship between  $g$  and

<sup>4</sup> It is not necessary for  $\mathcal{O}_1$  and  $\mathcal{O}_2$  to be distinct.

$m$  exists. Standard binary FCA is applied on this formal context to generate the associated formal concepts organized in a concept lattice. In the DrugBank example,  $G$  is the set of genes and  $M$  is the set of drugs. Considering  $g \in G$  and  $m \in M$ , we have  $(g, m) \in I$  if and only if a relationship between the gene  $g$  and the drug  $m$  exists in DrugBank.

## 5.2 Annotating the Concept Lattice with $\mathcal{O}_1$ and $\mathcal{O}_2$

Consider a formal concept  $(A, B)$  from the concept lattice.  $A \subseteq G$  is a set of individuals instantiating classes of  $\mathcal{O}_1$  and  $B \subseteq M$  is a set of individuals instantiating classes of  $\mathcal{O}_2$ . From the definition of the derivation operators in FCA, we know that every individual in  $A$  is in relationship with every individual in  $B$ . To find the most specific ontology classes involved as domain and range of the predicate, *two* annotations for formal concepts are defined (one per ontology) using the formalism defined in Subsection 3.2. We use two functions  $\delta_1$  and  $\delta_2$  to associate an individual with the set of the most specific classes from  $\mathcal{O}_1$  and  $\mathcal{O}_2$  that this individual instantiates.  $\delta_1$  is applied on individuals  $g \in A$  and  $\delta_2$  is applied on individuals  $m \in B$ .  $\sqcap_1$  and  $\sqcap_2$  are used to compute the similarity between descriptions of two individuals w.r.t. the two considered ontologies  $\mathcal{O}_1$  and  $\mathcal{O}_2$ . Finally, to compute the two annotations for each formal concept  $(A, B)$ , two derivation operators  $(\cdot)^{\circ_1} : 2^G \rightarrow 2^{\mathcal{C}(\mathcal{O}_1)}$  and  $(\cdot)^{\circ_2} : 2^M \rightarrow 2^{\mathcal{C}(\mathcal{O}_2)}$  are defined as follows:

$$A^{\circ_1} = \prod_1 \delta_1(g) \quad B^{\circ_2} = \prod_2 \delta_2(m)$$

It is noteworthy that  $(\cdot)^{\circ_2}$  is applied on the intent of the formal concept. As a result, each formal concept  $(A, B)$  is replaced by the annotated concept  $(A, B, A^{\circ_1}, B^{\circ_2})$  where  $A^{\circ_1}$  (resp.  $B^{\circ_2}$ ) is the set of the most specific classes of  $\mathcal{O}_1$  (resp.  $\mathcal{O}_2$ ) that all individuals in  $A$  (resp. in  $B$ ) instantiate.

## 5.3 Reading the Domain and Range of a Relation from the Annotated Lattice

Let us consider an annotated concept  $(A, B, A^{\circ_1}, B^{\circ_2})$ . Every individual in  $A$  is in a relationship with every individual in  $B$ . Furthermore,  $A^{\circ_1}$  is the set of the most specific classes of  $\mathcal{O}_1$  that all individuals in  $A$  instantiate. Similarly,  $B^{\circ_2}$  is the set of the most specific classes of  $\mathcal{O}_2$  that all individuals in  $B$  instantiate. Therefore, from this annotated concept, for the considered predicate, classes from  $A^{\circ_1}$  as domain are associated with classes from  $B^{\circ_2}$  as range.

Along the hierarchy of the lattice, the two annotations  $A^{\circ_1}$  and  $B^{\circ_2}$  behave in the opposite way. Indeed,  $A^{\circ_1}$ , computed on the extent, contains more specific classes when the number of individuals in the extent decreases, *i.e.*, when browsing the lattice top to bottom. On the contrary,  $B^{\circ_2}$ , computed on the intent, contains more specific classes when the number of individuals in the intent decreases, *i.e.*, when browsing the lattice bottom to top. Consequently, in annotated concepts at the top of the lattice, general classes of  $\mathcal{O}_1$  will be involved

in domains and specific classes of  $\mathcal{O}_2$  will be involved in ranges. In annotated concepts at the bottom of the lattice, specific classes of  $\mathcal{O}_1$  will act as domain and general classes of  $\mathcal{O}_2$  will act as ranges. Therefore, this structure could be of interest in an interactive setting with an expert. Indeed by browsing the lattice depending on her specific constraints on the domains or ranges to discover, the expert could leverage on the lattice order to obtain more general or specific classes involved.

## 6 Discussion

Regarding the suggestion of equivalence relationships between classes of two ontologies, the approach needs to be validated on a real dataset where mappings already exist. One main identified drawback is that the current annotation process works under the *Closed World Assumption*. Thus, mappings are suggested considering that all instantiations are correct and none is missing. As many datasets are under the *Open World Assumption*, the suggested mappings may only be applicable on the considered set of individuals but may not be applied to other sets of individuals. Therefore, there is a need to validate the suggested mappings with an expert. One next challenge would be to define a new derivation operator to compute an annotation working under the *Open World Assumption*.

The original lattice is built from the set of individuals and the classes of a reference ontology  $\mathcal{O}_{ref}$ . However, other features could be considered for this original classification, such as the relationships involving the individuals, similarly to our previous work [9]. The choice of the features to consider are of importance as they impact the original lattice, which is the the “pivot” structure from which equivalence relationships are suggested. One user can then choose the specific features to consider to guide the generated mappings. In the proposed approach, equivalence relationships are suggested by considering annotated concepts separately. Nevertheless, the subsumption relations between concepts could be considered to suggest mappings of the form of subsumption relations between classes of the two considered ontologies. Such setting could therefore be used to align and structure folksonomies of various users in a social network.

Finally, the suggestion of mappings may not be the only use case of interest for the proposed setting. Indeed, by annotating the original lattice with classes of different versions of the same ontology, concept drift could be highlighted. For example, a semantic change in the classes between two versions of an ontology would be indicated by discovering in the annotations that the same set of individuals does not instantiate the same classes between the two versions.

Regarding the use case of discovering classes frequently associated as domain and range of a predicate, we could also benefit from an experiment on a real dataset. In this setting, as previously mentioned, there is an issue in selecting the interesting domains and ranges. Additionally to the interactive discovery process previously mentioned, various metrics could be considered to highlight interesting annotated concepts. For example, notions of *support* or *confidence* based on the cardinal number of the extent and / or intent could be of interest



here. Similarly to the suggestion of equivalence relationships, this work does not benefit from the subsumption relations between formal concepts. Such relations could be used to build a hierarchical organization of associations of domains and ranges. However, as the two annotations behave in opposite ways, this hierarchy should not be read as an order.

## 7 Conclusion

In this paper, we combine Concept Annotation with the formalism of Pattern Structures. To illustrate our approach and its interest, we consider two use cases: suggesting equivalence relationships between classes of two ontologies and discovering classes frequently associated as domain and range of a predicate. The formalism of Pattern Structures is an advantage compared to our original work on Concept Annotation as it enables more complex annotations, using descriptions of objects and similarity operations. Moreover, we illustrate some other annotation possibilities: multiple annotations for one concept and annotations computed on the intent instead of the extent. As a result, starting from a concept lattice considered as a “pivot” structure, it is possible to obtain a complex structure representing or highlighting several relations between its components. The next challenges lies in experimenting our approach on real datasets and formalizing the properties of the annotation.

## Acknowledgments

This work is supported by the *PractiKPharma* project, founded by the French National Research Agency (ANR) under Grant No.: ANR-15-CE23-0028, and by the *Snowball* Inria Associate Team.

## References

1. Baader, F., Ganter, B., Sertkaya, B., Sattler, U.: Completing Description Logic Knowledge Bases Using Formal Concept Analysis. In: IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007. pp. 230–235 (2007), <http://ijcai.org/Proceedings/07/Papers/035.pdf>
2. Baader, F., Küsters, R., Molitor, R.: Computing least common subsumers in description logics with existential restrictions. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages. pp. 96–103 (1999), <http://ijcai.org/Proceedings/99-1/Papers/015.pdf>
3. Bendaoud, R., Napoli, A., Toussaint, Y.: Formal Concept Analysis: A Unified Framework for Building and Refining Ontologies. In: Knowledge Engineering: Practice and Patterns, 16th International Conference, EKAW 2008, Acirezza, Italy, September 29 - October 2, 2008. Proceedings. pp. 156–171 (2008). [https://doi.org/10.1007/978-3-540-87696-0\\_16](https://doi.org/10.1007/978-3-540-87696-0_16)

4. Coulet, A., Domenach, F., Kaytoue, M., Napoli, A.: Using Pattern Structures for Analyzing Ontology-Based Annotations of Biomedical Data. In: Formal Concept Analysis, 11th International Conference, ICFCA 2013, Dresden, Germany, May 21-24, 2013. Proceedings. pp. 76–91 (2013). [https://doi.org/10.1007/978-3-642-38317-5\\_5](https://doi.org/10.1007/978-3-642-38317-5_5)
5. Euzenat, J., Shvaiko, P.: *Ontology Matching*, Second Edition. Springer (2013)
6. Ganter, B., Kuznetsov, S.O.: Pattern Structures and Their Projections. In: Conceptual Structures: Broadening the Base, 9th International Conference on Conceptual Structures, ICCS 2001, Stanford, CA, USA, July 30-August 3, 2001, Proceedings. pp. 129–142 (2001). [https://doi.org/10.1007/3-540-44583-8\\_10](https://doi.org/10.1007/3-540-44583-8_10)
7. Ganter, B., Wille, R.: *Formal concept analysis - mathematical foundations*. Springer (1999)
8. Gruber, T.R.: A translation approach to portable ontology specifications. *Knowledge acquisition* **5**(2), 199–220 (1993)
9. Monnin, P., Lezoche, M., Napoli, A., Coulet, A.: Using Formal Concept Analysis for Checking the Structure of an Ontology in LOD: The Example of DBpedia. In: *Foundations of Intelligent Systems - 23rd International Symposium, ISMIS 2017, Warsaw, Poland, June 26-29, 2017, Proceedings*. pp. 674–683 (2017). [https://doi.org/10.1007/978-3-319-60438-1\\_66](https://doi.org/10.1007/978-3-319-60438-1_66)