A Proposal for an Interactive Ontology Design Process based on Formal Concept Analysis

Rokia Bendaoud, Amedeo Napoli, Yannick Toussaint

A proposal for an Interactive Ontology Design Process based on Formal Concept Analysis

Rokia BENDAOUD, Amedeo NAPOLI and Yannick TOUSSAINT
Campus Scientifique B.P. 239,
54506 Vandoeuvre-lès-Nancy Cedex, FRANCE

Abstract. Building a domain ontology usually requires several resources of different types, e.g. thesaurus, object taxonomies, terminologies, databases, sets of documents, etc. where objects are described in terms of attributes and relations with other objects. One important and hard problem is to be able to combine and merge knowledge units extracted from these different resources within the representation formalism supporting the ontology. The purpose of this paper is to show which kinds of resources can be taken as starting points for building an ontology, using FCA and its extension RCA. A real-world example in microbiology is proposed, detailing the interaction with domain experts during the ontology design process. Finally, an evaluation based on recall and precision gives an idea of the efficiency of the approach and points out several research perspectives.

Keywords. Ontology, FCA, RCA, Interaction with experts

1. Introduction

Ontologies are now widely introduced in the semantic web technology as they help software and human agents to communicate and to share domain knowledge [9,12]. In theory, an ontology is considered as an explicit specification of a domain conceptualization [7] represented within a formal language such as description logics (DL). In practice, despite of methodologies and methods for building ontologies [6] as well as concrete experiments [1], representing domain knowledge within a formal language such as description logics remains a very complex task using a manual approach: detecting inconsistencies in formal definitions of concepts may lead to a time-consuming and difficult restructuring of the whole ontology. One way of guiding the design task is to rely on an iterative ontology design process where the expert is only asked for very simple descriptions of the entities. In this way, as this is the case most of the time, the domain expert and the knowledge engineer are asociated for achieving the always complex task of real-world knowledge acquisition.

Building an ontology is an interactive process which requires several iterations before the expert agrees on the target ontology. One strategy, difficult to carry on

and to scale, would be for the expert to successively correct the target ontology according to his needs by directly changing the OWL code. However, if he wants to re-built the ontology by adding new resources or updates, the expert previous interactions are lost: they cannot be applied again to the OWL concepts that changed. A better way of considering interactions with the system is to ask the expert to adjust source material and to apply again the ontology design process. In fact, resources data are not modified but a preprocessing step has to be defined and acts as a filter. It enables the experts to perform operations on the source material.

Accordingly, one point of this paper is an iterative approach for building an ontology where the expert is asked to assign – or select into resources – objects, very simple attributes, and relations between two objects. These descriptions can be found in various resources having different types, e.g. thesaurus, vocabularies, dictionaries, sets of documents and databases. Starting with domain objects, a process that automates (in a certain way) the definition and design of concepts is of first importance. The FCA and RCA formalisms can guide the design of the ontology [10,3]. Then, concepts emerging from FCA and RCA can be encoded within a DL formalism. The evolutions of the ontology, *i.e.* addition, modification, and deletion, are not performed on the ontology itself but rather on the source material used to build the ontology. The source material for the ontology consists in a set of prepared data that will be the basis of the ontology design : for example, a binary table $\texttt{objects} \times \texttt{attributes}$ in FCA. Every time the ontology has to be changed, the source material is changed, the ontology design process is replayed and the target ontology is rebuilt.

Three main types of resources are distinguished in the following: a thesaurus, a database, and a set of documents. In a standard way, the thesaurus provides a set of hierarchically organized classes as `Klebsiella Pneumonae (Klebsiella P.)` is a `Proteobacteria`. The database and the set of documents provide a set of pairs $(\texttt{object}, \texttt{attribute})$ (attribute or property) and a set of triples $(\texttt{object}_\texttt{i}, \texttt{relation}, \texttt{object}_\texttt{j})$. For example, the class of `Helicobacter Pylory (Helicobacter P.)` bacteria can be described by pairs such as `(Helicobacter P., aerobic)`, `(Helicobacter P., negativeGram)` and `(Helicobacter P., spherical)`. The relation `Resist` whose co-domain includes ten families of antibiotics is defined by triples such as `(Helicobacter P., resist, Ciprofloxacin)`.

FCA and RCA are the processes on which is based the transformation between source materials towards the target ontology. One important idea on which relies the process is the existence of a "source" or "pivot" ontology extracted from the database or the set of documents, and then to extend the source ontology by progressively adding units extracted from the chosen resources. This source ontology is important with respect to the evaluation of the target ontology resulting from the whole design process. The addition of these units is based on the one hand on standard operations from FCA, such as apposition for example, and on the other hand on non standard operations such as RCA. A lattice with binary and relational attributes results from that process. Then, the elements in the target lattice –built thanks to FCA– can be represented within a knowledge representation language such as OWL. In this way, FCA is considered as the

"core" process in the design of the target ontology from a set of heterogeneous resources. Firstly, FCA and RCA as well take into account all elements included within an ontology, namely objects (or individuals), attributes, and relations, for building concept lattices. Secondly, the FCA framework provides operations to manage concept lattices, e.g. updating the lattice when the set of objects or the set of attributes is modified, merging or linking concept lattices. Finally, the resulting concept lattice can be transformed into a concept hierarchy within a description logic (DL) or an OWL concept hierarchy. A classifier can then be used for classification-based reasoning, e.g. answering queries. There are approaches similar to the present work but the novelty here lies in the articulation of the different operations for building up the target ontology.

An operational platform has been designed and an experiment in microbiology is detailed at the end of the paper to show the capabilities of the approach, the efficiency of an FCA-based transformation approach, and the usefulness of expert interactions with the system for reaching a consensus with respect to the target ontology.

The paper is organized as follows. The second section discusses requirements for designing an ontology from a set of heterogeneous resources. The third section introduces FCA and RCA, and the transformation process from a concept lattice to a target concept hierarchy within a DL-based framework. The fourth section presents interaction with experts in a real-world example for the design of a target ontology in microbiology. An evaluation of the ontology design process follows. Related and future work is examined at the end of the paper.

2. Merging simple descriptions to build an ontology

In this section, we analyze the basic objects and the resources that can be considered for building an ontology. For making precise every notion, the application domain chosen in this paper is microbiology. Three main kinds of basic objects are involved, namely genes, bacteria and antibiotics. The current problem is to build an ontology on the base of a collection of heterogeneous resources about resistance of bacteria to antibiotics by genes mutations. For bacteria, the following resources have been considered:

- The NCBI taxonomy (from the National Center for Biotechnology Information) includes 13380 species of bacteria.
- A collection of textual documents composed of 1244 abstracts has been selected by domain experts from PubMed (`http://www.ncbi.nlm.nih.gov/sites/entrez`), with a large collection of texts in the NCBI library.
- The pathogenic bacteria database (`http://bac.hs.med.kyoto-u.ac.jp/`).

For antibiotics, a concept lattice of ligands has been designed based on expert available knowledge (mainly involving chemical properties of antibiotics). For genes, the gene ontology[1] has been used.

[1] http://www.geneontology.org/

2.1. Three main types of object descriptors

Ontologies are usually not built from scratch and several kinds of resources can be used. Actually, the type of the resources does not matter as much as the type of information the resources include. In this paper, three main types of object descriptors are distinguished, (OD1) hierarchical links, (OD2) binary attributes, and (OD3) relational attributes (or binary relations),

(OD1). In an application domain, there are usually existing "source" hierarchies organizing domain objects, e.g. thesaurus or local ontologies from Swoogle[2]. Such hierarchies provide a global and structured view of the domain. In these hierarchies, a class denotes a set of objects and the relation between classes is set inclusion, while objects are instances of the class and all objects in a class are also in the superclasses. For example, `Klebsiella-pneumoniae` (or `Klebsiella-P.`) is a kind of `Proteobacteria`. Such classes can be compared to primitive concepts in description logics, as they do not have any explicit definition. In the context of microbiology, the NCBI taxonomy has played the role of source hierarchy.

(OD2). There are some resources such as databases where domain objects are described by means of a set of attributes. For example, `helicobacter pylori` has the `negativeGram` attribute (in the pathogenic bacteria database).

(OD3). Domain objects are related. Such relations occur in texts, but not exclusively. For example, the sentence "We have previously reported that a significant percentage (44%) of *isoniazid-resistant Mycobacterium tuberculosis* strains carry an arginine to leucine mutation in codon 463 (R463L) in the catalase-peroxidase gene (katG)." indicates that there exists a *resistance* relation from *Mycobacterium tuberculosis* to *isoniazid*. Such relation has been extracted from texts using GATE[3] [2]. It participates to the definition of classes of objects as well as attributes.

2.2. From a pivot ontology to a completed target ontology

The structure of the target ontology and its content has to take into account the three types of descriptors, *(OD1)*, *(OD2)*, and *(OD3)* introduced here-above: hierarchical links, attributes, and relations. Domain objects are grouped into a same class if and only if they share a given set of common attributes and relations. Both attributes and relations are necessary and sufficient conditions for defining a class of objects. For example, let us suppose that the `X` bacteria resists drug `D1`, the `Y` bacteria resists drug `D2`, and that `D1` and `D2` are drugs of the family `D`. In this context, `X` and `Y` can be grouped in the same class as they share the relation "resisting a drug from the class `D`". The resistance relation impacts on the definition of bacteria (here the domain of the relation). This shows in particular that attributes should be combined with relational attributes for forming richer and more precise definitions.

One main idea underlying the design of the target ontology is to rely on a "pivot" or "source" ontology, that will be progressively completed by the concepts

[2] http://swoogle.umbc.edu/

[3] `http://gate.ac.uk/`

extracted from the other resources. In the present framework, the NCBI taxonomy after being processed by FCA (as explained just after) has played the role of source ontology. The other resources that have been analyzed for completing the source ontology hold on genes, bacteria, and drugs.

The purposes of a target ontology depend in part of the type of queries expected to be asked. The structure and the content of the present target ontology should allow to ask three main types of queries.

- *(Q1).* Let $\mathtt{o}_1$ and $\mathtt{o}_2$ be two domain objects. Does it exist a class containing both objects or are these objects incompatible? What are the other objects in the common class. How is defined this common class?
- *(Q2).* Given a new object, say $\mathtt{x}$, that has been observed with some attributes and relations with other objects. What is the best and the right way of inserting this object in the ontology? Is there a class already available for this object or a new class has to be created?
- *(Q3).* What is the class of an object knowing the domain and/or the range of a relation. In particular, when $\mathtt{r}_1(\mathtt{o}_1,\mathtt{o}_2)$ and $\mathtt{o}_1$ is an instance of $\mathtt{C}_1 = \forall \mathtt{r}_1.\mathtt{A}_1$, then it can be inferred that $\mathtt{o}_2$ is an instance of $\mathtt{A}_1$.

3. Formal Concept Analysis

Formal Concept Analysis (FCA) and its extension Relational Concept Analysis (RCA) take into account the three main types of object descriptors discussed in Section 2. The FCA process builds concept lattices and provides various operations for managing concept lattices, in particular merging sets of objects or sets of attributes. RCA extends the scope of FCA by taking into account relational attributes. Moreover, the resulting concept lattice can be transformed into a concept hierarchy represented within the description logic formalism for allowing formal representation and reasoning.

3.1. Formal Concept Analysis

Formal concept analysis (FCA) [5] is a mathematical formalism allowing to derive a concept lattice from a formal context $\mathbb{K} = (G, M, I)$. FCA has been used for a number of purposes among which knowledge modeling, acquisition, and processing, lattice and ontology design, information retrieval, and data mining. In $\mathbb{K}$, G denotes a set of objects, M a set of attributes, and I a binary relation defined on the Cartesian product $G \times M$. In the binary table representing $I \subseteq G \times M$, the rows correspond to objects and the columns to attributes. The concept lattice is composed of *formal concepts* (or simply *concepts*) organized into a lattice by a partial ordering, i.e. a subsumption relation comparing concepts. A concept is a pair (A, B) where $A \subseteq G$, $B \subseteq M$, and A is the maximal set of objects sharing the whole set of attributes in B (and vice versa). In a concept (A, B), A is called the *extent* and B the *intent* of the concept. The concepts in a concept lattice are computed on the basis of a *Galois connection* defined by two derivation operators denoted by $'$:

$$' : A' = \{m \in M | \forall g \in A : (g, m) \in I\}$$
$$' : B' = \{g \in G | \forall m \in B : (g, m) \in I\}$$

A concept (A, B) verifies $A' = B$ and $B' = A$. The subsumption relation ($\sqsubseteq$) between a concept and a superconcept is defined as follows: $(A_1, B_1) \sqsubseteq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 (or\ B_2 \subseteq B_1)$. Relying on this subsumption relation $\sqsubseteq$, the set of all concepts extracted from a context $\mathbb{K} = (G, M, I)$ is organized within a complete lattice, called *concept lattice* and denoted by $\underline{\mathfrak{B}}(G, M, I)$.

The standard FCA process is able to deal with object descriptors of type *(OD1)* or *(OD2)*. Given a set of resources including such object descriptors, concept lattices provide a representation of the content of these resources. Then, the content of these resources can be merged using the FCA operation called *apposition*, as explained below.

	Proteobacteria	γproteobacteria	Actinobacteria	Bacilli
Helicobacter P.	X			
Klebsiella P.	X	X		
Mycobacterium S.			X	
Streptococcus P.				X
Klebsiella O.	X	X		

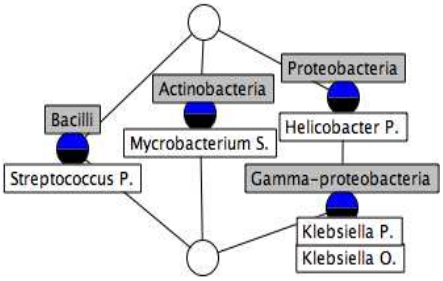

Figure 1. The context Bacteria from the database NCBI $\mathbb{K}_1 := (G, M_1, I_1)$ and the associated concept lattice.

Building a lattice from a hierarchy (OD1 object descriptor). Transforming a set of objects organized within a hierarchy –or described by hierarchical links– into a lattice is a straightforward operation. The formal context $\mathbb{K}_1 := (G, M_1, I_1)$ is defined as follows: G is the set of domain objects, M_1 is the set of classes of objects, and I_1 assigns to an object its class and all superclasses in the hierarchy. For example, the bacteria `Klebsiella P.` is classified in the NCBI hierarchical resource as a `GammaProteobacteria`, which in turn is a subclass of `proteobacteria`. Figure 1 shows the context associated to NCBI classification and the corresponding concept lattice.

Building a lattice from domain expert description of objects (OD2 object descriptor). A classification based on domain expert description of objects, i.e. involving *(OD2)* object descriptors, can be carried out as follows. A formal context $\mathbb{K}_2 := (G, M_2, I_2)$ is composed of a set G of objects, a set M_2 of attributes, and a relation $I_2 \subseteq G \times M_2$ where $I_2(g, m_2)$ states that g has the attribute m_2 (actually, the set G of objects is the same for context $\mathbb{K}_1$ and $\mathbb{K}_2$). Figure 2 shows an excerpt of such a context describing various bacteria, their attributes, and the corresponding concept lattice. In the present case, this concept lattice has been built for associating characteristics attributes to bacteria according to expert domain knowledge.

	spherical	sticks	negativeGram	positiveGram	aerobic	anaerobic
Helicobacter P.	×		×		×	
Klebsiella P.		×	×			×
Mycobacterium S.		×		×	×	
Streptococcus P.		×		×	×	
Klebsiella O.		×	×			×

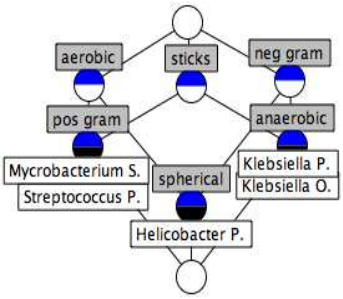

Figure 2. The context Bacteria based on expert knowledge $\mathbb{K}_2 = (G, M_2, I_2)$ and the associated concept lattice.

3.2. Merging two lattices with apposition in FCA

At this point, there are two contexts $\mathbb{K}_1 := (G, M_1, I_1)$ and $\mathbb{K}_2 := (G, M_2, I_2)$, with the same set of objects G and two distinct sets of attributes, M_1 and M_2 ($M_1 \cap M_2 = \emptyset$). The apposition operation is used in FCA for merging two contexts with the same set of objects and disjoint sets of attributes into a single context [5].

Definition 1 *Let* $\mathbb{K}_1 = (G_1, M_1, I_1)$ *and* $\mathbb{K}_2 = (G_2, M_2, I_2)$ *be two formal contexts. When* $G = G_1 = G_2$ *and* $M_1 \cap M_2 = \emptyset$, $\mathbb{K} := \mathbb{K}_1 | \mathbb{K}_2 := (G, M_1 \cup M_2, I_1 \cup I_2)$ *is the* apposition *of the two contexts* $\mathbb{K}_1$ *and* $\mathbb{K}_2$.

The two contexts are $\mathbb{K}_1 = (G, M_1, I_1)$ shown in Figure 1 and $\mathbb{K}_2 = (G, M_2, I_2)$ shown in Figure 2. In the apposition context $\mathbb{K} = (G, M, I)$, G is the set of objects –the same set for $\mathbb{K}_1$ and $\mathbb{K}_2$– $M := M_1 \cup M_2$ where M_1 is the set of attributes in $\mathbb{K}_1$ and M_2 is the set of domain attributes in $\mathbb{K}_2$, and $I := I_1 \cup I_2$. The resulting concept lattice is presented in Figure 3.

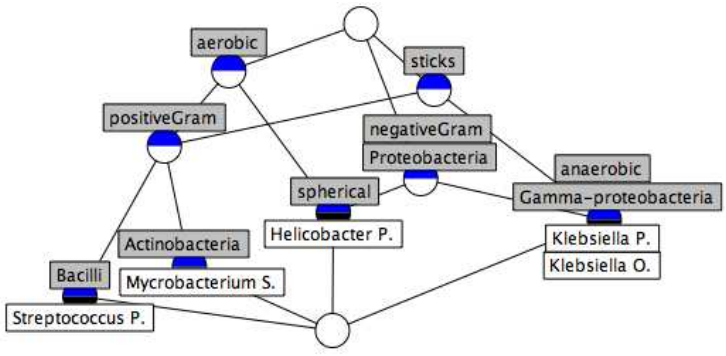

Figure 3. The concept lattice resulting from the apposition of contexts $\mathbb{K}_1$ and $\mathbb{K}_2$.

Table 1. The relation "Resist" between bacteria and antibiotics.

Resist				
	Clarithromycin	Ciprofloxacin	Cefotaxim	Macrolide
Helicobacter-P.		×		
Klebsiella-P.				×
Mycobacterium-S.			×	
Streptococcus-P.		×		
Klebsiella-O.	×			

3.3. Relational Concept Analysis

Relational Concept Analysis (RCA) [10] was introduced as an extension of FCA for taking into account relations between objects. A concept is then described with standard binary attributes and also with relational attributes. A relational attribute, say r, describes the relation existing between objects that are instances of a concept, say c_1, the domain of the r relation, with objects that are instances of another concept, say c_2, the range of r relation. RCA was already been used in a previous work on text mining and ontology design [2].

	FRB1	FRB3	ARB1	ARB2	HBA5	HBA10
Clarithromycin		×				×
Ciprofloxacin	×			×	×	
Cefotaxim		×	×			×
Macrolide			×			×

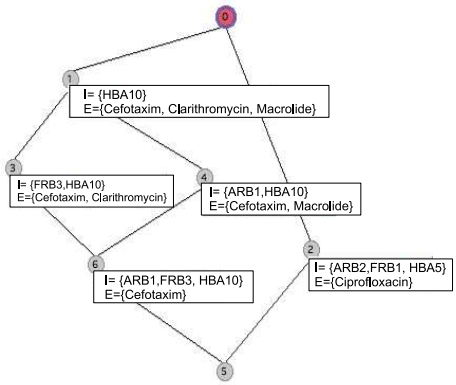

Figure 4. The context Antibiotics $\mathbb{K}_3 = (G_3, M_3, I_3)$ and the associated concept lattice.

Data in RCA are organized within a *relational context family* (RCF) composed of a set of contexts $\mathbb{K}_i = (G_i, M_i, I_i)$ and a set of relations $r_k \subseteq G_i \times G_j$. The sets G_i and G_j are the object sets of the contexts $\mathbb{K}_i$ and $\mathbb{K}_j$, called respectively the *domain* and the *range* of the relation r_k.

RCA uses the mechanism of *relational scaling* for defining relational attributes. For a relation, say $r : G_i \longrightarrow G_j$, linking objects from G_i to objects of G_j, a relational attribute is created and denoted by $r : c$, where c is concept in $\underline{\mathfrak{B}}(G_j, M_j, I_j)$. Then, for an object $g \in G_i$, the relational attribute $r : c$ character-

izes the "correlation" between g and $r(g) = h$ which is an instance of the concept $c = (X, Y)$ in $\underline{\mathfrak{B}}(G_j, M_j, I_j)$. Many levels of correlation can be considered such as the "existential correlation" –or existential scaling– where $r(g) \cap X \neq \emptyset$, and the "universal correlation" –or universal scaling– where $r(g) \subseteq X$. In the present work, only existential scaling is considered.

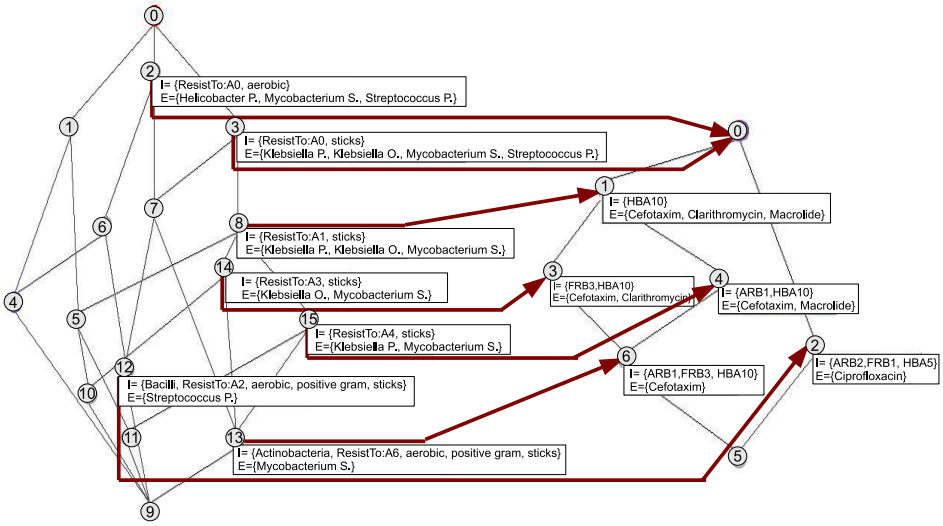

Figure 5. The lattice resulting from the RCA process applied to object descriptors of type *(OD3)*.

Let us consider the relation between bacteria and antibiotics, where the first context is given by context apposition in Figure 3 and the second context $\mathbb{K}_3 = (G_3, M_3, I_3)$ is given in Figure 4. The relation `Resist` between bacteria and antibiotics is given in Table 1. The application of RCA on the contexts of Figure 3 and Figure 4 produces the final concept lattice shown in Figure 5, where the relations explicitly computed by the RCA process are emphasized.

3.4. From concept lattice to DL formalism

The transformation of the final concept lattice resulting from RCA is based on a transformation into a DL knowledge base (KB) [10,11,8]. This transformation allows to introduce primitive and defined concepts, and thus to apply a DL-based reasoner for problem-solving and complex query answering. The target DL formalism is $\mathcal{FLE}$, that includes the constructors $\top$ (top), $\bot$ (bottom), $C \sqcap D$ (concept conjunction), $\forall r.C$ and $\exists r.C$ (universal and existential role quantifications). This set of constructors is large enough for representing all elements from the final concept lattice.

4. Interpretation and evaluation

4.1. Expert interaction with the system

The expert is invited to interpret the target ontology and to identify points in the ontology where there may be no agreement on the classification of objects or on the definition of classes. The reasons of these conflicts are: (1) there may be noise in resources or in the information extraction processes, (2) the expert is not satisfied with the target ontology and wants it to be more in accordance with his needs. In both case, the expert may apply elementary operations on the source material, and then run the FCA/RCA process for obtaining an updates version of the ontology.These operations depends on the object descriptors and are the following:

Operations on hierarchical link resources (OD1).

- *Adding a new class.* A new class is considered in the source hierarchy. This leads to add a new column to the formal context representing this hierarchy the. Then, expert has to assign to this new class the appropriate objects.
- *Changing the class of an object.* When changing the class of an object, the line describing the object in the formal context has to be modified: the new class and all its superclass have to be properly assigned to the object.
- *Deleting a class.* This operation was not used in this experiment. Deleting a class in the source hierarchy is equivalent to a deletion of a column in the formal context describing the resource.

Modifying attributes (OD2 or OD3). Quality of resources may depend on their form: database, text... For example, Natural Language Processing tools extracting information from texts are noisy when the linguistic level is too detailed compared to the ontological level. Some purely linguistically relevant information are deleted by the experts and some other may be introduced. The following operations can be used by experts:

- *Merging attributes.* This operations is relatd to synonymy in the texts. Expert may decide to merge the `positiveGram` with the `neutralGram` attribute for avoiding over-spliting classes in the target ontology.
- *Deleting an attribute for an object.* An attribute has been wrongly assigned to an object while extracting information from a resource; experts want to remove it. In the formal context describing this resource the cell (object,property) is changed to "blank".
- *Deleting an attribute for all objects.* The expert while interpreting the ontology observes that an attribute is not relevant. The column with this attribute in the formal context is simply deleted.
- *Adding an attribute to a set of objects.* The expert considers that an attribute is missing in a class. Either it is missing in the resources, either it has not been extracted (from texts). This operation is used for adding a column in the formal context and the attribute has to be assigned to the appropriate objects.

Operations on relational attributes are similar to operations on attributes. With this set of operations, the systel is able to meet the expert requests to converge towards the final ontology.

4.2. Expert interpretation

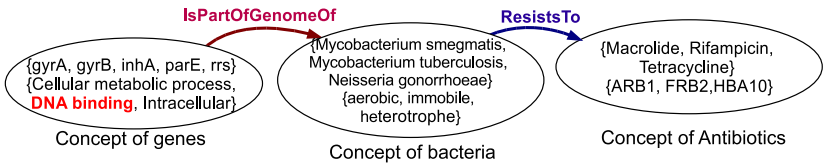

Figure 6. An example of an interpretation for the link between three classes : genes-bacteria-antibiotics

In this section, classes resulting from the lattices are presented and discussed. In the example of Figure 6, the expert found an explanation for the resistance of the set of bacteria `{Mycobacterium smegmatis, Mycobacterium tuberculosis, Neisseria gonorrhoeae}` to the set of antibiotics `{Macrolide, Rifampicin, Tetracycline}`. The explication is : the set of antibiotics `{Macrolide, Rifampicin, Tetracycline}` kill bacteria by detroying the DNA and the fact that the set of genes `{gyrA, gyrB, inhA, parE, rrs}` has the property of binding DNA allows bacteria to resist to antibiotics.

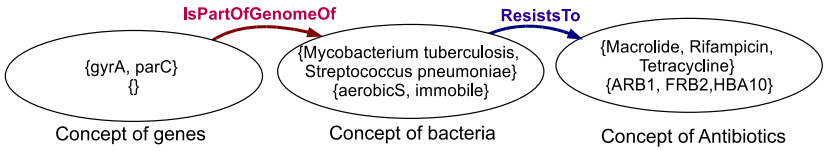

Figure 7. An other example of an interpretation for the link between three classes : genes-bacteria-antibiotics

In the second example on Figure 7, the set of genes `{gyrA, parC}` has not common attribute but the expert found this set interresting because it is knoown that there exists a strong relation between these two genes, and the first cannot be found without the second.

Another example is given by the concept `{Citrobacter freundii, Enterobacter aerogenes, Enterobacteriaceae, Escherichia coli, Pseudomonas aeruginosa, Salmonelle typhimurium, Serratia marcescens} {ResistTo:c4, ResistTo:c7, batonnet, gramNeg, hétérotrophe, mobile}`. The expert did not consider this class as interesting beacause these bacteria are different but there were no discriminant and characteritic attribute for separating these bacteria. One proposition of the expert was to add the attribute “activity Oxydase”.

5. Related work and Conclusion

In this paper, we have presented an original approach for building a target domain ontology in considering resources of different types, such as a thesaurus, term hierarchies, databases, and sets of documents.There are some work similar to the present one.

In [4], the authors use an approach which is able to acquire semantic knowledge from syntactic parsing and they use then FCA for building the concept hierarchy. Our approach deals with FCA, but uses in addition RCA and takes into account heterogeneous resources.

In [13], the authors propose to merge two ontologies for building a new one. The proposed method takes as input a set of documents. NLP techniques are used to capture two formal contexts encoding the relationships between documents and concepts in each ontology. This method combines the knowledge of the collection of texts and expert knowledge. This approach uses texts for merging and not for enriching the two ontologies.

In our framework, the resources are heterogeneous. Objects are described in terms of attributes and relations with other objects. Using FCA and its extension RCA, these different resources are transformed into source material and then represented as concept lattices. These concept lattices are used for completing a chosen reference concept lattice, that is the basis of the target ontology. This final concept lattice is transformed within a description logic formalism. Complex question-answering and classification-based reasoning can then be carried out using the classifier in the framework of description logics. A real-world example in microbiology has been detailed, showing the capabilities of the approach.

References

[1] N. Aussenac-Gilles, B. Biébow, and S. Szulman. Revisiting ontology design: A method based on corpus analysis. In Dieng R. and O. Corby, editors, *12th Internationnal Conference in Knowlegde Engineering and knowledge Management (EKAW'00)*, volume 1937, pages 172–188, 2000.

[2] R. Bendaoud, M. Rouane-Hacene, Y. Toussaint, B. Delecroix, and A. Napoli. Text-based ontology construction using relational concept analysis. In G. Flouris and M. d'Aquin, editors, *Proceedings of the International Workshop on Ontology Dynamics, Innsbruck (Austria)*, pages 55–68, 2007.

[3] R. Bendaoud, Y. Toussaint, and A. Napoli. Pactole: A methodology and a system for semi-automatically enriching an ontology from a collection of texts. In P. Eklund and O. Haemmerlé, editors, *Proceedings of the 16th International Conference on Conceptual Structures, ICCS 2008, Toulouse, France*, Lecture Notes in Computer Science 5113, pages 203–216, 2008.

[4] P. Cimiano, A. Hotho, and S. Staab. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research (JAIR)*, 24:305–339, 2005.

[5] B. Ganter and R. Wille. *Formal Concept Analysis, Mathematical Foundations*. Springer, 1999.

[6] A. Gómez-pérez, M. Fernández-Lòpez, and O. Corcho. *Ontological Engineering*. Springer Verlag, 2004.

[7] T. Gruber. Toward principales for the design of ontologies used for knowledge sharing. In Guarino N. and R. Poli, editors, *Formal Analysis in Conceptual Analysis and Knowledge Representation*, The Netherlands, 1993. Kluwer Academic.

[8] M. Huchard, A. Napoli, M. Rouane-Hacene, and P. Valtchev. Mining description logics concepts with relational concept analysis. In P. Brito, P. Bertrand, G. Cucumel, and F. De Carvalho, editors, *Selected Contributions in Data Analysis and Classification*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 259–270. Springer, Berlin, 2007.

[9] A. Maedche. *Ontologies Learning for the Semantic Web*. Springer, 2002.

[10] M. Rouane-Hacene, M. Huchard, A. Napoli, and P. Valtchev. A proposal for combining formal concept analysis and description logics for mining relational data. In S.O. Kuznetsov and S. Schmidt, editors, *Proceedings of the 5th International Conference on Formal Concept Analysis (ICFCA 2007), Clermont-Ferrand*, LNAI 4390, pages 51–65. Springer, Berlin, 2007.

[11] M. Rouane-Hacene, A. Napoli, P. Valtchev, Y. Toussaint, and R. Bendaoud. Ontology learning from text using relational concept analysis. In P. Kropf, M. Benyoucef, and H. Mili, editors, *International Conference on eTechnologies (MCETECH 08), Montréal*, pages 154–163. IEEE Computer Society, 2008.

[12] S. Staab and R. Studer, editors. *Handbook on Ontologies*. Springer, Berlin, 2004.

[13] G. Stumme and A. Maedche. Fca-merge: Bottom-up merging of ontologies. In *Proceedings of IJCAI'01, Seattle (WA)*, pages 225–234, 2001.