Formal Concept Analysis: A unified framework for building and refining ontologies

Rokia Bendaoud, Amedeo Napoli, and Yannick Toussaint

UMR 7503 LORIA, BP 239, 54506 Vandœuvre-lès-Nancy, FRANCE
{Rokia.Bendaoud,Amedeo.Napoli,Yannick.Toussaint}@loria.fr

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 an homogeneous formal representation (such as a description logic or OWL). The purpose of this article is to show which kinds of resources should be available for designing a real-world ontology in a given application domain, and then how Formal Concept Analysis and its extension - Relational Concept Analysis- can be used for materializing an associated ontology. This resulting target ontology can then be encoded within OWL or a description logic formalism, allowing classification-based reasoning. A real-world example in microbiology is detailed. Finally, an evaluation including tests on recall and precision shows how source resources can be completed with other existing domain resources using a semi-automatic analysis process.

1 Introduction

Ontologies are the backbone of Semantic Web as they help software and human agents to communicate and to share domain knowledge [1]. In theory, an ontology is considered as an explicit specification of a domain conceptualization [15]. In practice, an ontology may depend on various resources with different types, e.g. thesaurus, vocabularies or dictionaries, sets of documents, databases. Moreover, the web makes an increasing number of ontologies available for reuse. None of these ontologies can pretend to be complete but rather brings a specific point of view on a particular domain. Besides ontologies, resources of other types exist but are heterogeneous and most of the time disconnected. One important need in the framework of semantic web is to take advantage of all these types of resources. Thus, there is a need for integrating or "pushing" these various types of resources with the objective of knowledge sharing, updating, dissemination, communication, and being complete as much as possible with respect to a given domain. However, this wide range of resources should be interoperable and resource contents are represented within a common standard language such as, e.g. OWL[1] (as assumed within the framework of semantic import of modular ontologies in [8]), but this is not always the case.

[1] OWL : Web Ontology Language

Following this way, this paper aims at presenting a framework based on Formal Concept Analysis (FCA) for integrating and preparing various types of resources for allowing collaboration, interoperability, and the design of a domain ontology for problem-solving and reasoning. This is the role of Formal Concept Analysis to fill the gap between resources of various types and a "target ontology" encoded within the OWL language. The paper introduces a framework where FCA and its extension, Relational Concept Analysis (RCA), can be considered as integrating processes for resource integration, leading from a set of heterogeneous resources to a set of formal and homogeneous ontologies, and finally to a given target ontology.

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. 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 `firmicute` bacteria can be described by pairs, e.g. a set of attributes such as $\{\texttt{aerobic}, \texttt{negativeGram}, \texttt{spherical}\}$, and by triples such as the relation `ResistTo` whose co-domain includes ten families of antibiotics. These pairs and triples are taken into account within binary contexts for being processed by FCA and RCA for designing the final and integrated ontology. Moreover, pairs and triples will participate in defining a domain concept `C` with the help of necessary and sufficient conditions for testing the membership of an object `x` to the set of instances of `C`. As detailed later, pairs and triples also help in making explicit elements of implicit knowledge that are not directly accessible in the sole thesaurus. Meanwhile, the thesaurus –as understood here– plays a central role in the target ontology and for the domain expert (that is in charge of the interpretation of the extracted knowledge units for example). The thesaurus can be seen as a reference to which the target ontology can be compared. In this way, firstly, a class in the thesaurus may have been split into say two distinct classes, meaning that the attributes or relations observed in the other resources lead to the existence of these two classes: in this way, the elements of knowledge present in the original thesaurus have been made precise and completed. Secondly, two existing classes in the thesaurus may be merged into a more general class in the target ontology. An explanation may be that the two original classes in the thesaurus share a sufficient number of attribute for being identified in the new organization of the target ontology, meaning to some extent that the original distinction is not meaningful with respect to the resources examined during the process.

FCA and RCA are the processes on which is based the transformation between resources towards the target ontology. One important idea on which relies the process is the existence of a "source" or "pivot" ontology obtained from the thesaurus, and then to extend the source ontology by progressively adding units extracted from the resources under study. The addition of these units is based on the one hand on standard operations from FCA, such as apposition for example, but on the other hand on non standard operations such as RCA. Then, the

elements in the final concept lattice –built thanks to FCA– can be represented within the knowledge representation language OWL. In this way, FCA is considered as the "core" process in the design of the target ontology from a set of heterogeneous resources. This is the objective of this paper to explain how FCA and its extension RCA can be used for building, completing, and updating, a domain ontology. 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 lattices can be almost straightforwardly transformed into a concept hierarchy in a description logic (DL) $\mathcal{FLE}$ [5] or OWL concept hierarchy. A classifier can then be used for classification-based reasoning, e.g. answering queries. There are similar approaches but the novelty here lies in the articulation of the different operations for building up the target ontology. Moreover, an operational platform has been designed and detailed tests at the end of the paper show the capabilities of the approach and the efficiency of an FCA-based transformation approach.

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 concept hierarchy within a DL-based framework. The fourth Section presents a real-world example of the design of a target ontology from a set of heterogeneous resources in microbiology. An evaluation of the ontology design process follows. Related work is examined at the end of the paper.

2 Elements for building an ontology from heterogeneous resources

In this section, we analyze the basic objects and the associated resources that have to be considered for building an ontology in a given application domain. The domain chosen in this paper is microbiology, and two main kinds of basic objects are involved, i.e. bacteria and antibiotics. The problem is to build an ontology about resistance of bacteria to antibiotics on the base of a collection of heterogeneous resources. 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`), 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 (involving mainly chemical properties of antibiotics).

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 (or unary relations), and (OD3) relational attributes (or binary relations),

(OD1). In application domains, there are usually existing "source" hierarchies organizing domain objects, e.g. thesaurus, local ontologies from Swoogle [9]... 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 "leaves" (terminal nodes): all objects in a class are also in the superclasses. For example, `Klebsiella-pneumoniae` (or `Klebsiella-P.`) is a kind of `Protebacteria`. Such classes can be compared to primitive concepts in description logics, as they do not have an explicit definition.

In the context of microbiology, the NCBI taxonomy has played the role of the "source" or the "pivot" domain hierarchy.

(OD2). For other kinds of resources, e.g. databases, domain objects are described by means of a set of attributes. For example, `helicobacter pylori` has the `negativeGram` attribute (in the pathogenic bacteria database). However, objects are not assigned to a class nor embedded into a hierarchy.

(OD3). Domain objects may be related to other objects. Such relations occur in texts, but not exclusively. For example, a 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 type of relations has to participate to the definition of classes of objects as well as attributes.

The processing of textual resources. Given the texts and the databases listed above, attributes and relations between objects were extracted by the GATE system[2]. In the present framework, the use of GATE consists in two main operations. The first operation is an extraction of different entities of the domain, e.g. bacteria, antibiotics, etc. A second operation is the identification of relations existing in the analyzed texts, e.g. resistance, susceptibility, etc. For example, the analysis with respect to the resistance relation of a sentence such as "The genes conferring resistance to doxorubicin and daunorubicin in S. peucetius have been sequenced." Returns the tagged text shown below, used for describing a resistance relation between two objects, namely bacteria and antibiotics.

```
<Resistance>
   <Bacteria> S. peucetius </Bacteria>
   <Antibiotic> doxorubicin </Antibiotic>
```

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

```
   <Antibiotic> daunorubicin </Antibiotic>
</Resistance>
```

2.2 From a reference 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 as hierarchical links, attributes, and relations respectively. Domain objects are grouped into the same class if and only if they share a given set of common attributes and relations. Both properties and relations are necessary and sufficient conditions for defining such a class of objects. For example, in microbiology, let us suppose that the `X` bacteria resists drug `D1`, the bacteria `Y` resists drug `D2`, and `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 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.

In the present framework, the NCBI taxonomy after being processed by FCA (to be explained after) has played the role of reference ontology *(OD1)*. The other resources that were analyzed to complete this reference ontology were describing genes, bacteria, and drugs.

The purposes of a target ontology depend partially on the type of queries one expects to ask. In the present context, the structure and the content of the target ontology should allow asking three main types of queries.

- *(Q1)*. Let $\mathtt{o}_1$ and $\mathtt{o}_2$ be two domain objects. Does there exist a class containing both objects or are these objects incompatible? What are the other objects in this class? How is this class defined ?
- *(Q2)*. Given a new object, say `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? For example, 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, and in particular merging sets of objects or sets of attributes. RCA extends the scope of FCA for dealing with relational attributes. Moreover, the resulting concept lattice can be transformed into a concept hierarchy represented within the description logic formalism to allow formal representation and reasoning.

3.1 Formal Concept Analysis

Formal concept analysis (FCA) [3] 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 $'$:

$$A' := \{m \in M |\ gIm\ for\ all\ g \in A\}$$
$$B' := \{g \in G |\ gIm\ for\ all\ m \in B\}$$

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 types of 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.

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 organized into a hierarchy, 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 (in the domain of microbiology) as a γ`Proteobacteria`, 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

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

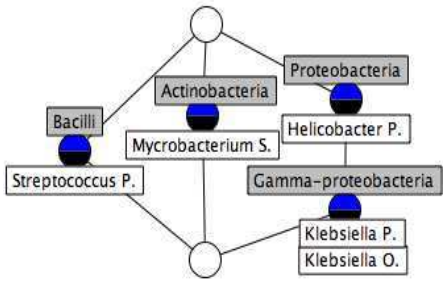

Fig. 1. The context Bacteria from the database NCBI $\mathbb{K}_1 := (G, M_1, I_1)$ and the associated concept lattice.

(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.

3.2 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. There exists an operation in FCA for merging these two contexts into a single one called *apposition* [3].

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 $\mathbb{K}_1$ –extracted from the NCBI hierarchy– 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.

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

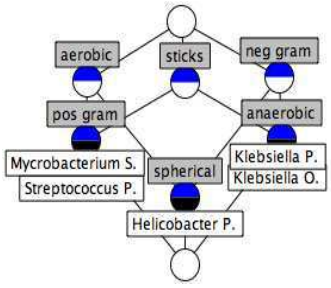

Fig. 2. The context Bacteria based on expert knowledge $\mathbb{K}_2 = (G, M_2, I_2)$ and the associated concept lattice.

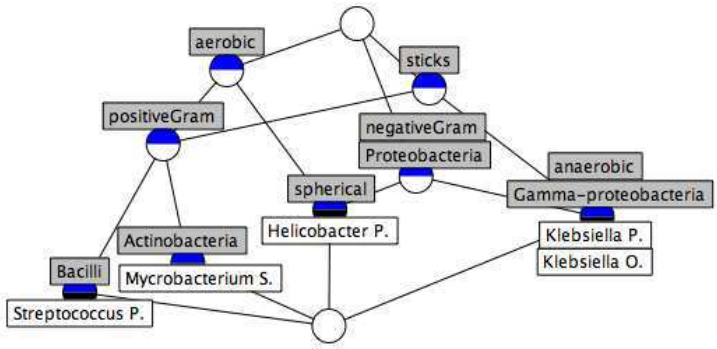

Fig. 3. The concept lattice resulting from the apposition of contexts $\mathbb{K}_1$ and $\mathbb{K}_2$.

3.3 Relational Concept Analysis

Relational Concept Analysis (RCA) [10] was introduced as an extension of FCA for taking into account relations between objects. In this way, a concept is described with standard binary attributes but 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 has already been used in a previous work in text mining and ontology design [13].

More precisely, 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 the so-called 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 $\mathbb{K}_j$. Then, for an object $g \in G_i$, the relational attribute $r : c$ characterizes the "correlation" between g and $r(g) = h$ which is an instance of the concept $C = (X, Y)$ in $\mathbb{K}_j$. Some correlations 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.

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 `ResistTo` between bacteria and antibiotics is given in Table 1. The application of the RCA process based on the concept lattices 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.

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

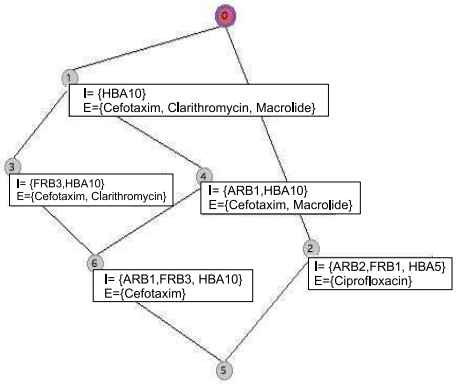

Fig. 4. The context Antibiotics $\mathbb{K}_3 = (G_3, M_3, I_3)$ and the associated concept lattice.

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

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

In more details, in Table 1, `Mycobacterium-S.` is related through `ResistTo` to `Cefotaxim` and `Streptococcus-P.` to `Ciprofloxacin`. Examining the lattice of antibiotics on Figure 4, it can be seen that `Cefotaxim` is in the extension of concepts `A0`, `A1`, `A3`, `A4`, and `A6`, while `Ciprofloxacin` is in the extension of concepts `A0` and `A2`. The relational attributes `ResistTo:A0`, `ResistTo:A1`, `ResistTo:A3`, `ResistTo:A4`, and `ResistTo:A6`, are associated to the object `Mycobacterium-S.`, while the relational attributes `ResistTo:A0` and `ResistTo:A2` are associated to `Streptococcus-P.` Then, a new concept lattice is built according to the extended context. At this point, as new concepts are been built, the lattice construction process is iterated and new relational attributes are associated to the bacteria objects whenever possible. If this is the case, the RCA process is iterated again. If this is not the case, this means that the fix-point of the RCA process has been reached and that the final concept lattice has been obtained. This final lattice is given on Figure 5 (lattice on the left). In particular, it can be seen that `Mycobacterium-S.` is in the extension of concept `C13` while `Streptococcus-P.` is in the extension of `C12`. All relational attributes intro-

duced above are respectively associated to `C13` and `C12` (attributes are inherited from upper concepts). It can be noticed that the identity of a concept is constant during the whole RCA process.

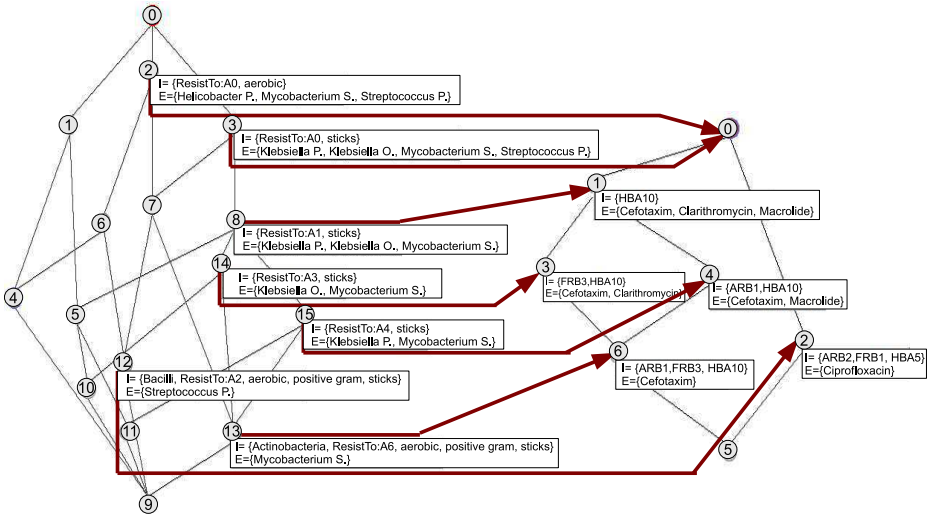

Fig. 5. The lattice resulting from the RCA process applied to object descriptors of type *(OD3)*.

4 From concept lattice to DL formalism

4.1 The representation of formal concepts into $\mathcal{FLE}$ concepts

The transformation of the final concept lattice resulting from RCA is based on a transformation, called τ, into a DL knowledge base (KB). The τ 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 $\mathcal{FLE}$ [5], 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. The profile of the τ transformation is the following:

$\tau : \underline{\mathfrak{B}}(G_f, M_f, I_f) \longrightarrow TBox \cup ABox$, where $\underline{\mathfrak{B}}(G_f, M_f, I_f)$ is the final concept lattice, TBox and ABox being the DL components on which the target ontology will be based. The concept lattice $\underline{\mathfrak{B}}(G_f, M_f, I_f)$ results from the FCA operations applied to the reference lattice –mainly appositions and relational scalings– which here is the concept lattice associated to the NCBI hierarchy (Figure 1). More precisely, the τ transformation works as follows:

- An attribute $m_1 \in M_1$, where $\underline{\mathfrak{B}}(G, M_1, I_1)$ of the concept lattice associated to the NCBI hierarchy, is transformed into an atomic –or primitive– concept in the TBox. This means that a class in the NCBI hierarchy is represented as an atomic concept, e.g. τ(`Proteobacteria`) = `Proteobacteria`.
- An attribute in a context distinct from $\underline{\mathfrak{B}}(G, M_2, I_2)$, e.g. in the Bacteria context associated to expert knowledge (see Figure 2), is transformed as a conceptual expression of the form $\exists$`m`.$\top$. For example, τ(`negativeGram`) = $\exists$`negativeGram`.$\top$.
- A relational attribute $r \in R$ is transformed in the TBox as an atomic role $\tau(r)$, e.g. τ(`ResistTo`) = `ResistTo`, i.e. an atomic role with the same name in the TBox.
- A formal concept $C = (X, Y)$ is transformed in the TBox as a defined concept formed by the conjunction of primitive concepts and existential role quantifications. For example, $\mathtt{C}_{12}$ = `Bacilli` $\sqcap$ $\exists$`sticks`.$\top$ $\sqcap$ $\exists$`aerobic`.$\top$ $\sqcap$ $\exists$`positiveGram`.$\top$ $\sqcap$ $\exists$`ResistTo`.$\mathtt{A}_2$ (see Figure 5).
  A subsumption relation between concepts is transformed as a general concept inclusion: the $C_1 \sqsubseteq C_2$ subsumption relation in the lattice becomes $\tau(C_1) \sqsubseteq \tau(C_2)$.
- An object $g \in G$ is transformed as an individual $\tau(g)$ in the ABox, e.g. `Staphylococcus aureus` becomes the individual τ(`Staphylococcus-aureus`).

Here are some examples of defined concepts:
$\mathtt{C}_2$ = $\exists$`aerobic`.$\top$ $\sqcap$ $\exists$`ResistTo`.$\mathtt{A}_2$
$\mathtt{C}_{12}$ = `Bacilli` $\sqcap$ $\exists$`sticks`.$\top$ $\sqcap$ $\exists$`aerobic`.$\top$ $\sqcap$ $\exists$`positiveGram`.$\top$ $\sqcap$ $\exists$`ResistTo`.$\mathtt{A}_2$
$\mathtt{C}_{13}$ = `Actinobacteria` $\sqcap$ $\exists$`sticks`.$\top$ $\sqcap$ $\exists$`aerobic`.$\top$ $\sqcap$ $\exists$`positiveGram`.$\top \sqcap \exists$`ResitTo`.$\mathtt{A}_6$

4.2 Reasoning within the DL formalism

The main reasoning operations that can be drawn are concept instantiation and concept subsumption, e.g. detecting the class of an individual –class stands here for concept extent or in DL terms as the set of instances of a concept– analyzing the range of a relation, or comparing concepts. Details for each reasoning operation are given below in the context of the microbiology example.

Instantiation consists in finding the class of an object (or individual). Let o_1 be an object with attributes {`a,b`} and relational attributes {$\mathtt{r}_1$`.`$\mathtt{A}_1$`,`$\mathtt{r}_2$`.`$\mathtt{A}_2$}, and belonging to classes {$\mathtt{C}_3$`,`$\mathtt{C}_4$} in the NCBI hierarchy. Then, the class of o_1 is the most general class `X` in the target ontology such that: `X` $\sqsubseteq$ $\mathtt{C}_3 \sqcap \mathtt{C}_4 \sqcap$ $\exists$`a`.$\top \sqcap \exists$`b`.$\top \sqcap \exists \mathtt{r}_1.\mathtt{A}_1 \sqcap \exists \mathtt{r}_2.\mathtt{A}_2$. This is a way of answering a question such as "What is the class of the object Streptococcus pneumoniae", whose attributes are {`aerobic,positiveGram,sticks`}, relational attributes are {`ResistTo:`$\mathtt{A}_2$}, and belonging to the class {`Bacilli`} in NCBI. According to the final lattice given in Figure 5, the answer is the concept $\mathtt{C}_{12}$.

A second task consists in determining whether two objects o_1 and o_2 have the same class. A simple way is to find the class of o_1, then the class of o_2,

and then to test whether the two classes are equivalent. For example, let us consider the objects `Klebsiella-O.` and `Streptococcus-P.`. `Klebsiella-O.` is an instance of the class $\mathtt{C}_{14}$ and `Streptococcus-P.` is an instance of the class $\mathtt{C}_{12}$ (see Figure 5). In this case, the fact that $\mathtt{C}_{14} \sqcap \mathtt{C}_{12} = \bot$ implies that both objects do not belong to the same class (or are incompatible).

Finally, the third task consists in detecting the class of an object knowing the domain or the range of a relation. Let us consider the instantiated relation $\mathtt{r}_1(\mathtt{o}_1,\mathtt{o}_2)$. When $\mathtt{o}_1$ is an instance of the class $\mathtt{C}_1 = \forall \mathtt{r}_1.\mathtt{A}_1$, it can be inferred that $\mathtt{o}_2$ is an instance of $\mathtt{A}_1$. When $\mathtt{o}_2$ is an instance of $\mathtt{A}_1$, it can be inferred that $\mathtt{o}_1$ is an instance of a class defined by an expression of the form either $\forall \mathtt{r}_1.\mathtt{A}_1$ or $\exists \mathtt{r}_1.\mathtt{A}_1$. For example, knowing that $\mathtt{ResistTo}(\mathtt{b}_1,\mathtt{a}_1)$ with $\mathtt{a}_1$ as an instance of `Ciprofloxacin`, it can be inferred that $\mathtt{b}_1$ is a bacteria, instance of concept $\mathtt{C}_{12}$ in Figure 5 (`Streptococcus-P.`).

5 Evaluation

There is no absolute and objective criteria to evaluate an ontology. Thus we decided to compare the target ontology to the NCBI thesaurus considered as the reference ontology.

We followed Maedche and Staab [1] approach, adapted by Cimiano et al. [12]. However, the representation of the lattice into DL following the function α makes the presentation of the evaluation quite different. This evaluation relies on similarity between sets of instances. First, for any class of the target ontology, its closest class in the reference ontology is computed.

Computing the closest class. The closest class is computed using the Euclidian distance on the set of instances. Let G be the set of objects, Ω_1 and Ω_2 be the two ontologies. For each class $C_1 \in \Omega_1$, and for each class $C_2 \in \Omega_2$, vectors V_{C_1} and V_{C_2} are defined as: $\forall g_k \in G$: if g_k is an instance of C_i then $V_{C_i}[g_k] = 1$ else $V_{C_i}[g_k] = 0$. Then:

$$Distance(V_{C_1}, V_{C_2}) = (\sum_{k=0}^{|G|}(V_{C_1}[g_k] - V_{C_2}[g_k])^2)^{1/2}$$

For the examples from figure 6: $Distance(\mathtt{sticks}, \mathtt{proteoacteria}) = \sqrt{3}$, and C_1 is the closest class of C_2 iff $\forall C \in \Omega_1 - \{C_1\}$ with $Distance(V_C, V_{C_2}) \geqslant (V_{C_1}, V_{C_2})$.

Computing precision and recall. We introduce three measures for ontology comparison. The precision for a given class $C_1 \in \Omega_1$ is computed with its closest class $C_2 \in \Omega_2$ as the proportion of instances from C_1 common to C_2. Recall is the proportion of instances from C_2 common to C_1.

$$Precision(C_1) = \frac{|C_1 \cap C_2|}{|C_1|}, \quad Recall(C_1) = \frac{|C_1 \cap C_2|}{|C_2|}$$

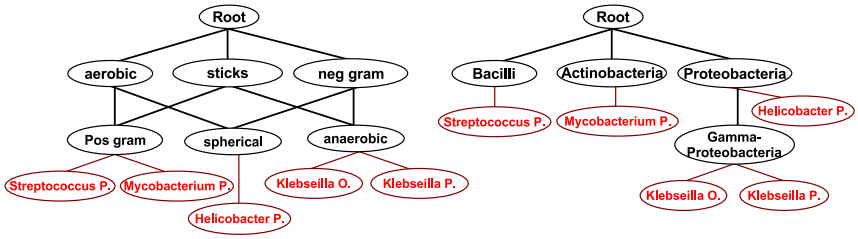

Fig. 6. Example of two ontologies, Ω_{target}(left) and $\Omega_{reference}$(right)

Table 2. The results of the evaluation of the FCA and RCA

	Number of classes	Number of properties	Number of relation	Precision	Recall	F-Measure	Defined concepts
FCA	58	13	0	76,52%	81,14%	78,76%	8/19
RCA	152	13	55	66,88%	82,07%	73,70%	12/19

Precision (resp. recall) is the average of precision (resp. recall) on all classes from the target ontology. F-measure is also defined as the harmonic mean of precision and recall. Let N be the number of classes in the target ontology and C_i a class in Ω_1:

$$P(\Omega_1, \Omega_2) = \frac{\sum_{i=1..N}(Precision(C_i))}{N}, \quad R(\Omega_1, \Omega_2) = \frac{\sum_{i=1..N}(Recall(C_i))}{N}$$

$$F(\Omega_1, \Omega_2) = \frac{2 * P(\Omega_1, \Omega_2).R(\Omega_1, \Omega_2)}{P(\Omega_1, \Omega_2) + R(\Omega_1, \Omega_2)}$$

For examples from figure 6, we have:
$P = \frac{1+1+1+1+\frac{1}{2}+1+1}{7} = 92,85\%$
$R = \frac{1+1+\frac{1}{3}+\frac{4}{5}+\frac{3}{5}+1}{7} = 81,90\%$

We have worked on two separate experiments: the first using only FCA (attributes) and the second using FCA + RCA (attributes + relational attributes). Table 2 presents the resulting precision, recall, and F-measure. This table presents also the number of classes with 100% precision and recall which FCA (and/or RCA) defines, i.e. gives necessary and sufficient conditions. RCA defines more classes than FCA (see Table 2). In following, we present an example of classes defined just with FCA and an example of classes defined by RCA.

FCA shows better precision and recall (see Table 2) than RCA. However, some classes need relations in their definitions, explaining why RCA shows a better number of defined classes. For example, let us consider the class C_9, i.e. `({Neisseria gonorrhoeae}, {Betaproteobacteria, Neisseria, Proteobacteria})`. Using FCA does not allow to find a closest class with precision and recall of 100%. Insted, using RCA, allows to

build the class C_{150} that has exactly the same set of instances ({Neisseria gonorrhoeae}, with precision and recall of 100%.

6 Related work

In [4], the authors present a cooperative machine learning system called ASIUM, which is able to acquire semantic knowledge from syntactic parsing. The system ASIUM successively aggregates clusters to form new concepts and the hierarchies of concepts from the ontology. The ASIUM approach differs from our approach because the former is not based on the same classification approach, does not work with heterogeneous resources, and does not try to complete ontologies for building a target ontology. In [12], the authors use an approach similar to the preceding approach, but they use the FCA for building the concept hierarchy. Regarding the work of Cimiano et al., our approach involves the use of FCA and RCA as well as taking into account heterogeneous resources.

The extraction of relational attributes allows a better definition of concepts. In [7], the authors propose the system "Lexical Navigation" for extracting the (non hierarchical) relations. Their idea is to use a lexical network containing domain-specific vocabularies and relationships that are automatically extracted from a collection of documents. In the same way, the work in [11] proposes to use a learning method to extract syntactic patterns. This method extracts manually relations between terms from texts and searches to generalize the terms and the relation between these terms. In comparison to our method, the two preceding methods try to cluster terms with a different classification approach, taking into account relations in texts but without a systematic approach as FCA and RCA. The facts of dealing with heterogeneous resources and with a final target ontology are also different.

Another approach described in [2] consists in extracting association rules from a collection of texts and keeping the rules having a given support and frequency. The objectives could be compared but the classification methods used in this work and ours are quite different.

In [6], 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. Comparing with our approach, the approach of Stumme et al. uses the texts for merging and not for enriching the two ontologies. The authors in [14] propose to enrich an existing ontology using on-line glossaries. They use natural language definitions of each class and convert them into formal definitions (OWL), compliant with the core ontology property specifications. Then, this method needs an existent core ontology for adding the transverse relations that is not our case.

7 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. In these resources, objects are described in terms of attributes and relations with other objects. Using the FCA process and its extension RCA, these different resources can be represented as concept lattices. These concept lattices are used to complete a chosen reference concept lattice, that will be the basis of the target ontology. Then, 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 that the approach is fully operational.

In this paper, only a part of the available and potential knowledge implicitly lying in the different resources has been extracted for analyzing the phenomenon of bacteria resisting to antibiotics. In future work, we plan to extend the target ontology by extracting other objects that are of importance, e.g. genes, codons, etc., and, as well, other relations, which include composition and spatial relations between bacteria, genes, and other biological actors present in the texts. In this way, a more precise representation of the content of texts will be designed and used for characterizing texts in microbiology. Finally, such a characterization could be used for comparing, classifying, and computing similarities between texts on the basis of their contents. This could also lead to sophisticated kinds of reasoning on texts, e.g. case-based reasoning and adaptation of a texts.

References

1. Maedche A. *Ontologies Learning for the Semantic Web.* Springer, 2002.
2. Maedche A. and Staab S. Discovering conceptual relation from text. In *14th European Conference on Artifical Intelligence (ECAI'00)*, pages 321–325, Berlin, Germany, 2000.
3. Ganter B. and Wille R. *Formal Concept Analysis, Mathematical Foundations.* Springer, 1999.
4. Faure D. and Nedellec C. A corpus-based conceptual clustering method for verb frames and ontology acquisition. In *Workshop on Adapting lexical and corpus resources to sublanguages and applications (LREC'98)*, 1998.
5. Baader F., Calvanese D., McGuinness D.L., Nardi D., and Patel-Schneider P.F., editors. *The description logic handbook: theory, implementation, and applications.* Cambridge University Press, New York, NY, USA, 2003.
6. Stumme G. and Maedche A. Fca-merge: Bottom-up merging of ontologies. In *International Joint Conference on Artificial Intelligence (IJCAI'01)*, pages 225–234, 2001.
7. Cooper J.W. and Byrd R.J. Lexical navigation: Visually prompted query expansion and refinement. In *2nd International Conference on Digital Libraries (DL'97)*, pages 237–246, 1997.
8. Pan J.Z., Serafini L., and Zhao Y. Semantic import: An approach for partial ontology reuse. In *1st International Workshop on Modular Ontologies (WoMO'06) In ISWC 2006*, 2006.
9. Ding L., Finin T.W., Joshi A., Pan R., Scott Cost R., Peng Y., Reddivari P., Doshi V., and Sachs J. Swoogle: a search and metadata engine for the semantic web. In Grossman D., Gravano L., Zhai C., Herzog O., and Evans D.A., editors,

International Conference on Information and Knowledge Management (CIKM'04), pages 652–659. ACM, 2004.
10. Rouane-Hacene M., Huchard M., Napoli A., and Valtchev P. A proposal for combining formal concept analysis and description logics for mining relational data. In Kuznetsov S.O and Schmidt S., editors, *Proceedings of the 5th International Conference on Formal Concept Analysis (ICFCA 2007), Clermont-Ferrand*, LNAI 4390, pages 51–65. Springer, Berlin, 2007.
11. Aussenac-Gilles N., Biébow B., and Szulman S. 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.
12. Cimiano P., Hotho A., and Staab S. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research (JAIR)*, 24:305–339, 2005.
13. Bendaoud R., Rouane-Hacene M., Toussaint Y., Delecroix B., and Napoli A. Text-based ontology construction using relational concept analysis. In Flouris G. and d'Aquin M., editors, *Proceedings of the International Workshop on Ontology Dynamics, Innsbruck (Austria)*, pages 55–68, 2007.
14. Navigli R. and Velardi P. Ontology enrichment through automatic semantic annotation of on-line glossaries. In Staab S. and Svátek V., editors, *15th Internationnal Conference in Knowlegde Engineering and knowledge Management (EKAW'06)*, volume 4248, pages 126–140, Podebrady, Czech Republic, 2006. Springer.
15. Gruber T.R. 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.