An FCA-based Solution for Ontology Mediation

Olivier Curé and Robert Jeansoulin
Université Paris Est, Institut Gaspard Monge,
77454 Marne la Vallée Cedex 2, France
{olivier.cure, robert.jeansoulin}@univ-paris-est.fr

Received 24 February 2009; Accepted 22 March 2009

In this paper, we present an ontology mediation solution based on the methods frequently used in Formal Concept Analysis. Our approach of mediation is based on the existence of instances associated to two source ontologies, then we can generate concepts in a new ontology if and only if they share the same extent. Hence our approach creates a merged ontology which captures the knowledge of these two source ontologies. The main contributions of this work are (i) to enable the creation of concepts not originally in the source ontologies, (ii) to propose a solution to label these emerging concepts and finally (iii) to optimize the resulting ontology by eliminating redundant or non pertinent concepts. Another contribution of this work is to emphasize that several forms of mediated ontology can be defined based on the relaxation of certain criteria produced from our method. The solution that we propose for tackling these issues is an automatic solution, meaning that it does not require the intervention of the end-user, excepting for the definition of the common set of ontology instances.

Categories and Subject Descriptors: Formal Methods **[Decision Sciences]**

General Terms: Ontology, Mediation

Additional Key Words and Phrases: Formal Concept Analysis, Merging, Alignment

1. INTRODUCTION

The trends we are witnessing in the evolution of Information Technology (IT) emphasize the preponderance of knowledge as opposed to data and information [Ackoff 1989]. A direct consequence of this situation consists in the developement of emerging information systems which are able to cope with the different ways to represent, maintain and query this knowledge. A particularly interesting and promising representation solution corresponds to ontologies.

First of all, we would like to clarify the kind of ontologies we are studying. In [McGuinness 2003], the author presents a spectrum of definitions of ontologies, ranging from controlled vocabularies to expressive logic-based and declarative formalisms. In

this work, we are interested in the common fragment of these formalisms corresponding to the concept hierarchy, i.e. a partial order of the ontology's concepts. In the remaining of this paper, we refer to the notions containing such a fragment, e.g. term hierarchies, taxonomies, classifications or logic-based representations, as an ontology.

It is important to stress the plethora of sources of such ontologies in IT and the World Wide Web (W3) in particular. In fact the W3, and especially the Semantic Web, is a distributed environment where ontologies excel. This is partly emphasized by the prevalence of this notion in the architecture proposed in [Berners-Lee et al. 2001] for the architecture of this next generation web. This situation motivated the W3C to order and supervise the development of ontology related standards, such as RDF(S), OWL and SPARQL. Many tools are also widely available, e.g reasoners like Pellet, editors like Protégé, triple stores like Sesame, programming frameworks and APIs like HP's Jena and the OWLAPI, etc..

The number of openly available ontologies represented using these standards is steadily increasing, and so is the number of applications using them, e.g. social networks, linked data, geographical information systems, medicine applications, etc.. For instance, the medical domain is a very active field of development with large, standardized and structured ontologies being produced (SNOMED CT, semantic network of UMLS, Galen, ATC in pharmacology, etc.). Another example is e-commerce where companies use ontologies to share information and to guide their customers through their Web sites.

With so many ontologies being produced, it is inevitable that some of their elements overlap. In order to support ontology interoperability, it is required that these ontologies are semantically related. Thus ontology mediation [Ehrig 2006] becomes a main concern. Ontology mediation enables to share data between heterogeneous knowledge bases, and allows applications to reuse data from different knowledge bases. Ontology mediation is generally considered to take one of the two following distinguished forms:

- Ontology mapping, where the correspondences between elements of two ontologies are stored separately from the ontologies. The correspondences are generally represented using axioms formulated in a peculiar mapping language.
- Ontology merging, which consists in creating a new ontology from the union of source ontologies. The merged ontology is supposed to capture all the knowledge of the sources.

Ontology mediation is an active research field where many kinds of solutions have been proposed: schema-based, instance-based, machine learning-inspired, hybrid approaches; see [Kalfoglou and Schorlemmer 2003] and [Euzenat and Shvaiko 2007] for surveys of ontology mapping and matching.

In this paper, we propose a solution for ontology mediation based on the techniques of Formal Concept Analysis (FCA) [Ganter and Wille 1999]. FCA algorithms are machine learning techniques that enable the creation of a common structure, which may reveal some associations between elements of the two original structures. We consider structures that are represented as ontologies. Thus it requires that some elements from both ontologies can be attached to a same observable item. The processing of our FCA-based algorithms provides a merged ontology. However, the

merged concepts are either exactly matched or linked by a subsumption relation. This enables to retrieve mapping axioms from our solution's results and thus to generate an alignment. Hence, we consider our solution to be an ontology mediation tool although we concentrate in this paper on its merging aspect.

The adoption of FCA as a technique to merge ontologies is motivated by its ability to discover and position new concepts in the merged ontology. In fact, given two partially ordered sets, FCA methods are able to define a new (merged) ordered set which may contain some nodes that were not present in the original structures. Moreover it is able to place them properly and accurately in the subsumption hierarchy of this new structure. That is a new concept will be placed as a direct subconcept of its most specific superconcept.

As said earlier, we consider as FCA structures a large common fragment of ontologies available today. Whereas most ontology merging solutions discover mappings between elements of both ontologies, they generally have difficulties to discover new concepts resulting from the merging process. Moreover, we show that this method is automatic, except for the objects extraction phase.

The solution we propose extends existing FCA-based systems for ontology merging in the following way: (i) we provide a method to create concepts not originally in the source ontologies, (ii) we provide labels to these newly created concepts, based on the labels of the implied source concepts and finally (iii) we optimize the resulting ontology by eliminating redundant and non-pertinent concepts. The step (i) is the classical approach named *ontology alignment* in FCA literature. In general, ontology mediation solutions can not handle concepts resulting from the fusion of ontologies. The steps (ii) and (iii) are an extension of this alignment, providing a basis for a possible interpretation, and an automated ranking allowing to choose between the new aligned concepts. Step (ii) is particularly useful for end-users of the merged ontology since we provide a meaningful label to the new concepts. Finally, with step (iii), we remove redundant and non-pertinent concepts of the merged ontology. Again, this is a very interesting features for ontology users.

A fourth contribution of this work is to enable the definition of several forms of mediated ontologies. We differentiate these ontologies in terms of the concepts that they contain and their subsumption hierarchy. This differentiation is performed using criteria that we defined using our FCA-based solution. This solution is automatic, up to the selection and setting of some measures presented in Section 4, and tackles the main challenges encountered in ontology merging, i.e. reflecting all the correspondences and discrepancies between the source ontologies.

The paper is organized as follows: in Section 2, we present some basic notions about FCA and the logical formalism we are using to represent the ontologies, namely Description Logics (DL). In Section 3, we present our FCA-based solution which involves several steps: concept generation, redundancy elimination and label generation. In Section 4, we propose solutions to refine the resulting ontology and present the criteria which serve to produce the different forms of merged ontologies: the dependencies and restrictions we are exploiting in our work. Section 5 relates our work with existing systems in ontology mediation and collaborations between FCA methods and DLs. Section 6 provides a discussion and concludes this paper.

2. BACKGROUND

2.1 Formal Concept Analysis

The ontology mediation solution that we propose, uses the methods of FCA. Intuitively, this means that we mediate two conceptual structures in a context consisting of a set of objects, a set of attributes (for each conceptual structure), and a set of correspondences between objects and attributes. FCA is based on the notion of a *formal context.*

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

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

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

The hierarchy of formal concepts is formalized by:

$$(A_1, B_1) \leq (A_2, B_2) \iff A_1 \subsetneq A_2 \text{ and } B_1 \subsetneq B_2$$

The concept lattice of $\mathcal{K}$ is the set of all its formal concepts with the partial order that we have represented witht the $\leq$ symbol.

Galois connections are related to the idea of order and play an important role in lattice theory. Let $(P, \preceq)$ and $(Q, \preceq)$ be two partially ordered sets (poset). A Galois connection between P and Q is a pair of mappings (Φ, Ψ) such that Φ: $P \to Q$, Ψ: $Q \to P$ and:

- $x \preceq x'$ implies $\Phi(x) \succeq \Phi(x')$,
- $y \preceq y'$ implies $\Psi(y) \succeq \Psi(y')$,
- $x \preceq \Psi(\Phi(x))$ and $y \Psi(\Phi(y))$

The hierarchy of formal concepts obeys the mathematical axioms defining a lattice, and is called a concept lattice since the relation between the sets of objects and attributes is a Galois connection.

2.2 Ontologies and Description logics

In Section 1, we insisted on the fact that several notions of ontologies exist and are exploited in IT. Our FCA-based solution to ontology merging exploits the fragment of concept hierachy that is commonly encountered in ontologies.

In order to represent this fragment, we are using a Description Logics (DLs) [Baader et al. 2003] formalism. This family of knowledge representation formalisms allows to reason over domain knowledge, in a formal and well-understood way. Central DL notions are concepts (unary predicates) and roles (binary predicates). A key notion in DLs is the separation of the terminological (or intensional) knowledge, called a TBox, to the assertional (or extensional) knowledge, called the ABox. The TBox is generally considered to be the ontology. Together, a TBox and a ABox represent a

Knowledge Base (KB), denoted $KB = \langle hTBox, ABoxi \rangle$.

Several reasoning services can be provided by a DL-system. A fundamental one, which is particularly important in our approach, corresponds to *concept subsumption*. This service, usually written $TBox \models C \sqsubseteq D$, can be defined as follows: Given a TBox T and two classes C an D, verify whether the interpretation of C is a subset of the interpretation of D in every model of T.

In this paper, we do not consider concept expressions that are found in expressive DLs and leave this aspect to future publications. Adopting this approach implies that we only consider atomic concepts and their hierarchies. This enables our approach to generalize to ontologies which take the form of term hierarchies, classifications and taxonomies. This has the advantage of widening the set of candidate ontologies applying to our solution, for instance in the medical or geographical domains. Finally, another advantage is that these simple and inexpressive ontologies can be found embedded in many practical databases, thus offering access to possibly large sets of instances. This aspect of accessing large datasets is particularly important in our approach.

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

3. ONTOLOGY MEDIATION USING FCA

Let consider two applications that manipulate data, about a common domain, e.g. geography or medicine. Each application uses independent ontologies to represent the concepts related to its specific data. In order to enable the exchange of data from one application to the other, it is necessary to discover correspondances between elements of their ontologies, i.e. their DL concepts. Figure 1 proposes a graphical representation of some simple ontologies, where lattice nodes *A*, *A*1, *A*2, *B*, *C*, *C*1, *C*2, *D* correspond to DL concepts, *Top* is the most general concept of these ontologies and *Bottom* to the empty concept. These ontologies will serve as the driving example all along this paper.

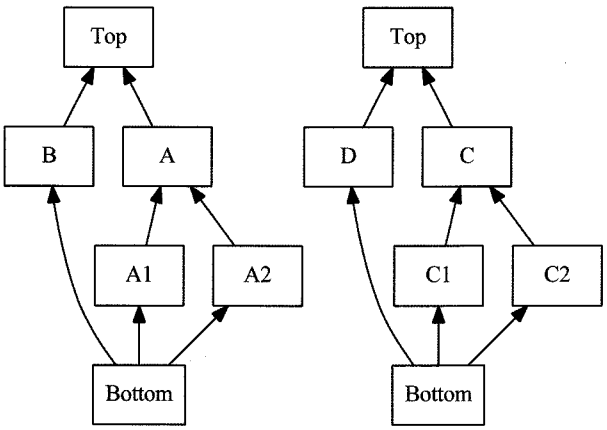


Figure 1. Source Ontologies to Align.

3.1 Accessing Instances of DL Concepts

We consider DL knowledge bases with non-empty ABoxes. Given the fact that both *KB*s share a common domain, it is likely that some individuals are simultaneously instances of both source ontologies.

One can ask how easily can individuals be retrieved from DL ABoxes. In fact, the infatuation surrounding the Semantic Web motivates several research teams to study cooperations between the domains of databases and knowledge bases represented in a DL. For instance, the Ontology-Based Data Access (OBDA) approach [Poggi et al. 2008] proposes to store the individuals of the ABox in a relational database and to represent the schema of this database in a DL TBox. Also tackling this same objective, the team supporting the Pellet reasoner, one of the most popular OWL reasoner, recently released `OWLgres` which is being defined by their creators as a 'scalable reasoner for OWL2'. A main objective of this tool is to provide a conjunctive query answering service using SPARQL and the performance properties of relational database management systems. Finally, [Hustadt et al. 2004] proposes an approach tailored to enable efficient Abox reasoning. This is performed by translating SHIQ DLs to disjunctive datalog and thus to apply practically successful deductive database optimization techniques.

Using one of these approaches, the set of observed objects may be retrieved from existing relational database instances using already existing FCA tools adapted to this technology, e.g. ToscanaJ [Becker and Correia 2004].

These instances leverage the identification of similar concepts. The discovery of these individuals usually requires interactions with a domain expert and can be considered:

- simple if the instances are identified by identical constant values, e.g. national drug identifiers in a pharmacology database,
- complex if the identification requires the exploitation of reference reconciliation [Dong et al. 2005], record linkage [Fellegi and Sunter 1969] and [Elfeky et al. 2002] or object identification [Peng Lim et al. 1993] tools.

We do not concentrate on the exploitation of these tools and invite interested readers to study the cited literature. Also, we are aware that the size of the instance dataset has an important impact on the computational efficiency of our solution. In the rest of this paper, we also ignore this issue and consider that a set of relevant instances is provided and is processed effectively by our algorithms.

The "mapping" we propose between both ontologies can be represented by a *matrix*. In Table I, we present an extract of a matrix of our running example. In this matrix, the rows correspond to the objects of $\mathcal{K}$, i.e. common instances of the KB's ABox, and are identified by integer values from 1 to 9 in our example. The columns correspond to FCA attributes of $\mathcal{K}$, i.e. concept names of the two TBox. In the same table, we present, side by side, the formal concepts coming from our two ontologies, i.e. $A1$, $A2$, A, B from Ontology 1, and $C1$, $C2$, C, D from Ontology 2.

3.2 Processing the Concepts of the Merged Ontology

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

Table I. Sample Dataset for Our Ontology Alignment Example.

	A1	A2	A	B	C1	C2	C	D
1	x		x		x		x	
2	x		x		x		x	
3		x	x		x	x		
4		x	x			x	x	
5		x	x					x
6				x	x		x	
7				x				x
8				x				x
9	x		x					x

ontologies:

- first, for each row, mark the columns where a specific instance is observed, e.g. the object on line 1 is an instance of the $A1$ and $C1$ concepts. Thus ABoxes information are used in this step.
- then, complete the row with the transitive closure of the subsumption relation between ontology concepts, e.g.: line 1 must be also marked for DL concepts A and C, as respective ontologies state that: $A1 \sqsubseteq A$ and $C1 \sqsubseteq C$. Here, the concept hierarchy of TBoxes are exploited.

Using Table I data, and the Galois connection method [Davey and Priestley 2002], we can generate the lattice of Figure 2. In this lattice, an arrow represents a subsumption relationship (leaving from the subconcept and pointing to the superconcept) and each node contains the following two sets:

- a set of objects identified by the integer values of the first column of our matrix. We denote this fragment of a node as the *extension fragment.*
- a set of DL concepts identified by the concept labels of our source ontologies. We denote this fragment of a node as the *intension fragment.*

In the next section, we consider the meaning of these concepts and as a first

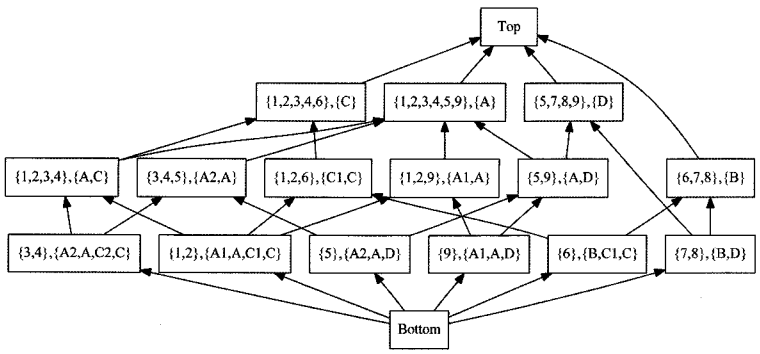


Figure 2. Galois Connection Lattice.

answering set, we propose a solution that provides them with labels.

3.3 Non-Redundant Naming of Mediated Concepts

A goal of our approach is to generate labels for the nodes of the merged lattice. Dependencies to and from these nodes are the main source of information for setting their labels. Given the number of dependencies for a given node, the generated name can be unnecessarily large. In this work, we consider the usability aspect of a generated merged ontology. A major issue for users of merged ontologies is to understand the semantics of the ontology concepts. Concept labels are a direct and easy conveyor of the semantics of these concepts. Hence, whenever new concepts are introduced by a merge operation, it is essential to provide them with relevant labels. In order to support this label generation, we introduce the notion of *label set minimality* which supports the creation of optimum and short labels for the lattice nodes.

Definition 3. Given a node N in the Galois connection lattice and a set of labels L contained in its intension fragment. We consider that L is minimal for N if and only if there is no L' for N such that $|L'|<|L|$, where $|L|$ denotes the size of L.

In the next section, we will propose an operator which enables to process minimal label sets. Also, given the previous definition, it is easy to demonstrate that several different minimal label sets may hold for a given node. We want to consider only the minimal label sets that satisfy the concept inclusion axioms (e.g. $C \sqsubseteq D$) of our (merged) ontology.

First, we consider that the extension fragment of a node of our lattice is not useful for its naming. Thus, we remove it from all the nodes of the lattice, and only concept names remain (the intension fragment). Nevertheless, in Section 4, we use extension fragments in order to calculate statistics.

Then, we can also optimize the labels used by the concept of the merged ontology. Due to the lattice structure obtained by applying the Galois connection method, we can proceed by using a top-down navigation, i.e. starting from the top concept (Top), on the concepts of the merged ontology. Basically, this algorithm (*optimize-Label*) proceeds as follows: for a given concept C of the lattice, it computes all its children c (line 1) and checks if the label used to characterize C is used in the label collection for c (line 2). If this the case, this label is removed from the label of c (line 3) otherwise the labels of c remain unchanged. Finally, the method is applied recursively to each concept c until all concepts are processed (line 5).

Algorithm 1	optimizeLabel (Concept C)
1	FOR EACH child c of C DO
2	IF label(C) ∈ label(c) THEN
3	remove label(C) from label(c)
4	END IF
5	optimizeLabel(c)
6	END DO

Processing this algorithm on our running example, we obtain Figure 3 where lattice nodes contain a single set, corresponding to concept names from some of the original ontologies or an empty set. Several kinds of nodes, in terms of the size of a name set, can be generated with this method. Basically, it is important to distinguish between the following three kinds of nodes:

- a singleton: a name of a concept from either original ontology, because it can be distinguished from any of its successors by this specific name, e.g. this is the case for the {C2} lattice node.
- an empty set, denoted by a variable (*_nx*), because it can not be directly distinguished from any of its possible successors. We have 7 such nodes in Figure 3. In section 3.4, we propose a labeling solution for these nodes. These nodes correspond to the new concepts of the merged ontology.
- a set of several names, all belonging to a given ontology, because the mediation based on the given two ABoxes, has not been able to split names. Indeed, it is as if the two names are glued together in a single concept name. We also consider this compound name as an *atomic* name (or label). In Section 4, we present such a situation and argue for the equivalence of the concepts corresponding to these labels.

All atomic labels are maintained in the resulting merged ontology but we need to find a labeling solution for the concepts of the second situation, i.e. those with 'empty' labels.

3.4 Label Generation for Unnamed Mediated Concepts

This section proposes a solution to the generation of labels for the 'empty' nodes. In our running example, we have identified seven such nodes which are displayed in Figure 3.

We propose a method based on breadth first search of the lattice and start from its most general concept (Top). The algorithm named *labelEmptyNode* exploits a FIFO

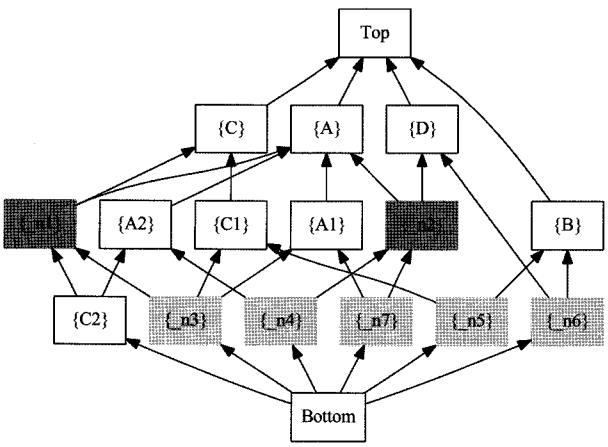


Figure 3. Galois Connection Lattice with 'empty' Nodes (Colored).

queue which proposes the methods *enqueue* and *dequeue* to add (respectively remove) an element to (resp. from) the queue.

The algorithm exploits a double marking of lattice nodes: one, denoted *mark*, to mark processed nodes and another one, denoted *markEmpty*, to mark 'empty' nodes. In this algorithm, we denote by the *intFrag* the intension fragment of an 'empty' node which contain an empty string prior to processing.

Before getting into the details of our algorithm, we need to emphasize on our need to handle a precise form of embedding in compound labels. Hence it is necessary to introduce a new operator, which we denote with the following symbol: $\oplus$. Intuitively, this operator is a binary relation between DL concept labels of the source ontologies. That is whenever two nodes, which we denote with α and β, are associated in the resulting lattice, we write $\alpha \oplus \beta$.

Thus the label generated for an 'empty' node may correspond to the association of several $\oplus$ operations denoted by $\oplus_{i=1}^{n}$ and defined as follows:

$\oplus_{i=1}^{n} \alpha_i = \alpha_1 \oplus ... \oplus \alpha_n$ where $\alpha_1,..,\alpha_n$ are all successors of the 'empty' node.

The algorithm works as follows: in lines (1,2,3), we create an empty queue, mark the 'Top' node and add it to the queue. Then until the queue is not empty (line 4) we remove the first element of the queue (line 5) and search for all its children (line 6). In a first step, the method searches if the element is not already (marked) in the queue (line 8), marks it (line 9) and add it to the queue (line 10). Then we check if this child is an 'empty' node (one of the _n1 to _n7 in our example) if this is the case, we mark it as being an 'empty' node (line 13) and update its intension fragment (line 14) taking care of embedding using brackets "( )".

Algorithm 2	labelEmptyNode (node N)
1	q = create queue
2	mark N
3	queue N to q
4	WHILE q $\neq \emptyset$ DO
5	x = dequeue q
6	WHILE x has child DO
7	z = next child e
8	IF notMarked z THEN
9	mark z
10	queue z to q
11	END IF
12	IF z is an 'empty' node THEN
13	markEmpty z
14	intFrag of z = intFrag of $\oplus$ (label of x)
14	END IF
15	END WHILE
16	END WHILE

After processing this method, we can clean the labels of 'empty' nodes by removing the inner most brackets.

Example 1 In the lattice presented in Figure 3, the nodes identified by values _n1 to _n7 have respective labels: "$A \oplus C$", "$A \oplus D$", "$(C \oplus A) \oplus C1 \oplus A1$", "$A2(A \oplus D)$", "$B \oplus C1$", "$B \oplus D$" and "$(A \oplus D) \oplus A1$".

The semantics we associate to this operator has two flavors. Concerning the set of individuals denoted by a node, i.e. a concept in our ontology, the $\oplus$ operator is equivalent to a conjunction. That is in a Tarski style semantics, we can write $(C \oplus D)^{\mathcal{I}} = C^{\mathcal{I}} \wedge D^{\mathcal{I}}$, where $\mathcal{I}$ denote an interpretation function over the domain of discourse of the ontology. Concerning the label generation of a given node, the $\oplus$ operator corresponds to a string concatenation on which several inference operation can be performed (see rules $R1$ and $R2$ in this section).

In the sequel, let α, β, γ and δ, be atomic concept labels. We consider non atomic (or compound) labels to be composed of atomic ones with $\oplus$ operations and brackets, e.g. $\alpha \oplus \beta$ is non atomic in $(\alpha \oplus \beta) \oplus \gamma$. A special attention has to be given to the properties of the $\oplus$ operator. Indeed, it does not have the associativity property, i.e. $(\alpha \oplus \beta) \oplus \gamma \neq \alpha \oplus (\alpha \oplus \gamma)$, but is commutative, i.e. $(\alpha \oplus \beta) \oplus \gamma = \gamma \oplus (\alpha \oplus \beta)$.

3.5. Minization of the Generated Labels

The labels generated using this technique are still bearing a part of redundancy (see below). Hence, we propose a solution that reduces the size of the generated labels, by reducing the redundancy that they contain. We exploit the DL concept hierarchy with its set of concept inclusion axioms (i.e. $C \sqsubseteq D$), to design derivation rules, to be applied to the $\oplus$ operator.

We first show that no inference rule can be applied to reduce the length of labels composed of atomic labels only : $\alpha \oplus \beta \oplus \gamma$. This is due to the structure of the Galois connection lattice generated and the *optimizeLabel* algorithm which does not integrate the transitive closure of the inheritance relationship. This means that our lattice can not contain a node with label $\alpha \oplus \beta$ with $\alpha \sqsubseteq \beta$. Inductively, this fact holds for $\oplus_{i=1}^{n} \alpha_i$ as long as α_i are atomic labels.

It is possible to apply some inference rules whenever we are in the presence of at least one non atomic label in a node. We distinguish several situations for node labels: they are formed of (i) non atomic labels (e.g. $(\alpha \oplus \beta)$) and atomic labels or (ii) non atomic labels only. We thus propose two inference rules to handle each situation:

$$(R_1)\ \frac{(\alpha \oplus \beta) \oplus \gamma,\ \gamma \sqsubseteq \alpha}{\beta \oplus \gamma}$$

$$(R2)\ \frac{(\alpha \oplus \beta) \oplus (\gamma \oplus \beta),\ \gamma \sqsubseteq \alpha}{\gamma \oplus \beta \oplus \delta}$$

Intuitively, the rule R1 states that if a node _n presents a label $(\alpha \oplus \beta) \oplus \gamma$ and the axiom $\gamma \sqsubseteq \alpha$ holds in the generated lattice, then it is possible to simplify _n's label into $\beta \oplus \gamma$.

Example 2 Suppose we have a node n whose generated label consists of: (*Scientist* $\oplus$ *Student*) $\oplus$ *GraduateStudent*. This means that _n inherits from a node _n' whose label is (*Scientist* $\oplus$ *Student*) and its interpretation contains all students in science, and also inherits from _n" whose label is *GraduateStudent*. Applying rule R1, reduces

the label of _n to *GraduateStudent* $\oplus$ Scientist as the ontology contains the *GraduateStudent* $\sqsubseteq$ *Student axiom.*

The rule R2 works similarly to R1 but handles situations where only compound labels are present in the node label.

Example 3 Suppose that our node _n has label (*ComputerScience* $\oplus$ *Student*) $\oplus$ (*Math* $\oplus$ *GraduateStudent*) and we consider that our merged ontology contains the axiom *GraduateStudent* $\sqsubseteq$ *Student*. Then by applying rule R2, we obtain the label *GraduateStudent* $\oplus$ *Math* $\oplus$ *ComputerScience* which is obvious from the context.

These rules show that the introduction and elimination of brackets needs to be handled with attention as they represent the introduction of generated labels at other levels of the lattice.

The application of these rules aims to produce "minimal labels". This notion is related to the fact that two concepts differ by a minimal set of changes (insertions or deletions of atomic concept labels) with respect to set inclusion.

Definition 4. Let ψ and ϕ be two concepts, we say: $\psi \subset \phi$ if and only if the set of atomic concepts of ψ is strictly included in the set of atomic concepts of ϕ.

Hence, a label is minimal if it contains the fewest number of atomic concepts.

Definition 5. Given a node label ψ directly computed from its successors in the Galois lattice, a new label ϕ for this node is minimal if and only if:
- it is computed with the derivation rules R1 and R2,
- $\phi \subset \psi$,
- there is no ϕ' such that $\phi' \subset \phi \subset \psi$.

Example 4 In our ontology example, the 'empty' node identified with node _$n3$ corresponds to this situation. Its label is first set to "$(C \oplus A) \oplus C1 \oplus A1$". To reduce the length of this label, we use the following dependencies:
- concept inclusion axiom $A1 \sqsubseteq A$ from Ontology 1.
- concept inclusion axiom $C1 \sqsubseteq C$ from Ontology 2.

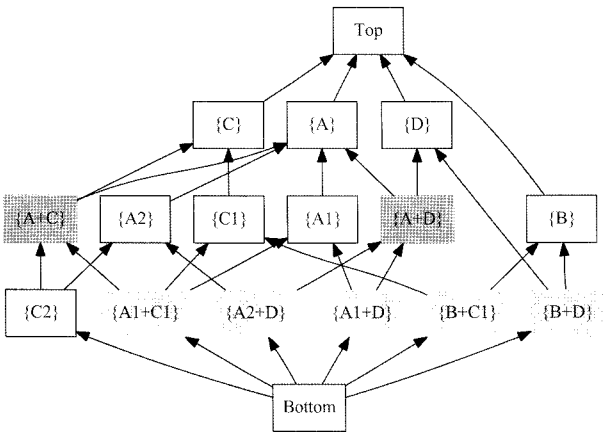


Figure 4. Galois Connection Lattice with Generated Labels.

Then, using R1, we can derive, in two steps, that $(C \oplus A) \oplus C1 \oplus A1 \equiv A1 \oplus C1$. Respectively we can deduce for $\_n4$ and $\_n7$: $A2 \oplus D$ and $A1 \oplus D$.

Figure 4 shows the mediated ontology resulting from application of FCA.

4. ONTOLOGY REFINEMENTS

The goal of this section is to examine what kind of confidence we can bring to the nodes of the mediated ontology, and to examine if this confidence can be graded over the new concepts, according to the information that we may extract from our knowledge. First, we identify the sources of uncertainty in the generation of our merged ontology. Then, we present some heuristics to measure the strength of some concepts, when it is possible. Finally, we propose a solution to rank the concepts in the merged ontology.

4.1 Dealing with Uncertainty

The two original ontologies may each have their own quality assessment method, or none, which can affect the confidence on TBox (generally high), or on ABox (generallymore questionable, and not uniform). But when considering the mediated (or merged) ontology, we know that the new concepts, computed by the FCA algorithm, rely on two TBoxes, on two ABoxes, and on the matrix that represented the mapping. This situation may cause some errors to occur at the different levels of our resulting merged ontology. This is basically a machine learning method, with all the inherent uncertainty belonging to such methods.

The problem with FCA, as with any machine learning method, is that datasets, or ABoxes, may contain errors. This problem somehow generalizes the problem of *integrity constraints* in relational databases (RDB), defined in terms of relations and dependencies [Kanellakis 1990]. Whenever a RDB is updated, dependencies can be checked: on success, the RDB instance is modified, otherwise the update is rejected. A study [Motik et al. 2007] shows how hard it is to use integrity constraints in DLs: one major difficulty is that DL assumes *open world*, while RDB relies on *closed-world assumption.*

Hence it may be very useful to detect 'wrong' ABox assertions, using another mechanism than checking integrity constraints. If the ABox contains errors, so do the input matrix, then the generated lattice may contain nodes which will later be transformed into wrong DL concepts in the mediated ontology. In order to filter the DL concepts we can generate with our FCA-based solution, we present several confidence measures in the next section.

4.2 Measuring Concept Confidence

We present three measures, namely Support, Derivation and Lattice Position, to assess to quality of the merged ontology.

4.2.1 *Support.* The notion of Support corresponds to a frequency measure based on the idea that values which co-occur together frequently have more evidence to justify that they are correlated and hence are more interesting.

Definition 6. We define the support of an 'empty' node _n in our Gallois connection

lattice as:

$$S\_n = \frac{\text{number of objects g} \in \mathcal{K} \text{ involved in the formal concept of \_n}}{\text{total number of objects in the formal context } \mathcal{K}}$$

This solution can also be exploited when a row in our matrix represents a cluster of objects and we associate the number of objects of this cluster to each row.

Example 5 With Table 1 and Figure 2, the support can be easily computed, e.g. the total number of objects on Table 1 is 9 and for the lattice concept identified with _n1, i.e. $\{A \oplus C\}$, the number of objects is 4. Thus the support of node _n1 denoted $S_{\_n1} = \frac{4}{9} = 0.44$.

The assumption behind the exploitation of the support measure is that the generation of wrong concepts can be avoided by setting a threshold below which concepts are not created. So far, we have only considered to detect wrong $\mathcal{K}$ objects to avoid producing incoherent DL concepts in the mediated ontology. But this detection can also serve to repair the ABoxes of the source DL ontologies.

4.2.2 *Derivation.* In order to provide a confidence value to the labels that have been generated, we consider a measure based on the number of derivations ($R1$ and $R2$) that have been performed in the process of generating the merged ontology. Intuitively, we consider that using the concept inclusion axioms of our source ontologies reinforces the belief that we should have on a generated label. This intuition expresses the confidence one has on the TBox information compared to the ABox information.

Definition 7. Let $|\phi|_R$ be the number of R1 and R2 derivations needed to obtain a label $\{\phi\}$: this number measures the confidence of the concept ϕ.

Example 6 The confidence value for node _n3 is $|A1 \oplus C1|_R = 2$.

We consider that the higher the value of $|\phi|_R$, the better is the confidence of the existence of this concept in the merged ontology. This assumption is due to the fact that the derivation measure is based on concept inclusion axioms, thus TBox related, defined in the source ontologies.

It is also interesting to note that all 'empty' nodes _n with $|\_n|_R = 0$ correspond to

Table II. Sample Dataset for the Label Differentiation Example.

	Support	*Derivation*	*Position*
_n1	4/9	0	non-leaf
_n2	2/9	0	non-leaf
_n3	2/9	2	leaf
_n4	1/9	1	leaf
_n5	1/9	0	leaf
_n6	2/9	0	leaf
_n7	1/9	1	leaf

a simple $\oplus$ operation on atomic concepts.

4.2.3 *Lattice position.* Finally, we consider that the position an 'empty' node _n has in the generated lattice has an important impact on the resulting merged ontology. That is removing any of the nodes _n3, _n4, _n5, _n6 or _n7 does not have the same impact on the lattice of Figure 4 as removing either _n1 or _n2. We define the lattice position, denoted Position, as follows.

Definition 8. For an 'empty' node _n, the position is:

$$Position(\_n) = \begin{cases} non-leaf & \exists \_n' Bottom \sqsubseteq \_n' \sqsubseteq \_n \\ leaf & \text{otherwise} \end{cases}$$

Example 7 $Position(\_n1) = non - leaf$ since $C2 \sqsubseteq \_n1$ and $Position(\_n3) = leaf$ since there is no nodes between _n3 and Bottom in the lattice of Figure 4.

4.3 Ranking Concepts in the Mediated Ontology

We now study how the concepts that we have created and labeled with our method can be trusted in our system. We use the measures that we have just defined and summarize the results over our running example in Table II. This table presents nodes _n1 to _n7 and their corresponding measure values.

The notion of differentiation of the mediated ontology corresponds to the fact that different merged ontologies can be produced via our method. These ontologies differ in terms of the number of DL concepts they contain and their concept subsumption hierarchy.

Given a merged ontology, it is possible to modify it with the following methods:

- modifying the dataset represented in the matrix and computing a new merged ontology using our method. We denote this approach as an *object level* modification of the merged ontology. This approach may be motivated by the discovery that some tuples of the matrix are incorrect.
- directly modifying the merged ontology by removing or introducing concepts. We denote this approach as a *concept level* modification of the merged ontology. An end-user may select this approach knowing that some concepts are inconsistent or missing from the merged ontology.

In Figure 4, we have already seen the most complete and complex lattice we can obtain from our running example. So starting from this lattice, the introduction of some new DL concepts in the merged ontology means that the dataset of the matrix is incomplete. Note that a complete repairing may involve to complete the source databases with missing tuples or $\mathcal{K}$ attributes. Concerning the elimination of DL concepts from the merged ontology, we argue that our three measures can help to safely remove some set of DL concepts.

We consider that the Support measure can be used at both object level and concept level. For instance, removing, from the matrix, objects involved in the lowest Support value (i.e. with value $\frac{1}{9}$ in Table II) yields the new merged ontology of Figure 5. In this figure, $A = C$ means that the DL concepts A and C are equivalent, denoted in $A \equiv C$ in DL formalism meaning that $A \sqsubseteq C$ and $C \sqsubseteq A$ (this corresponds to case (1)

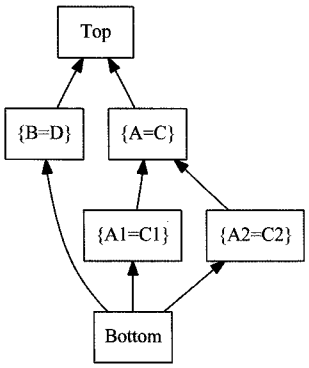


Figure 5. Galois Connection Obtained Removing Support Values of 1/9.

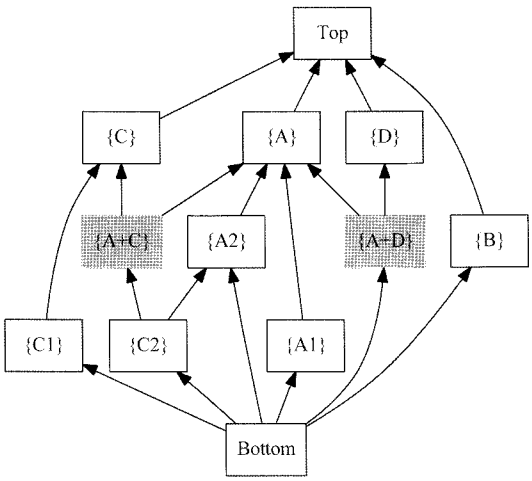


Figure 6. Galois Connection Obtained by Removing 'leaf' Positions.

in Section 3.4).

Position and Derivation are used at the object level. For instance, in Figure 6, we present a new merged ontology processed from Figure 4 by removing the DL concepts at the leaf position.

5. RELATED WORK

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

In the literature, two distinct approaches in ontology merging have been distinguished. In the first approach, the merged ontology captures all the knowledge of the source ontologies and replaces them. An example of such a system is presented in [Noy and Musen 2000] with the PROMPT tool. In the second approach the source ontologies are not replaced by the merged, but rather a so-called 'bridge ontology' is created. The bridge ontology imports the original ontologies and defines the

correspondences using axioms which are called "bridge axioms". An example of such an approach is the Ontomerge solution which has been described in [Dou et al. 2002].

The most relevant work related to our solution is the FCA-merge system [Stumme and Maedche 2001]. It uses instances of ontology classes to exploit an FCA algorithm, hence a first step consists in instance extraction from source ontologies. Then the FCA-merge system produces a lattice of concepts which relates concepts from the source ontologies. This new concept lattice is then handed to the domain expert in order to interactively generate a final merged ontology. Thus we can consider FCA-merge to be a semi-automatic solution while our solution aims to generate the merged ontology automatically. So the main differences are that FCA-merge is unable to propose concepts emerging from the fusion of the source ontologies and does not propose a label generation solution. Also, without the help of domain experts, the FCA-merge system is not able to refine the merged ontology.

Another interesting system is the GLUE system [Doan et al. 2002] which uses machine learning techniques to discover mappings between ontologies. Thus this project relates to the mapping aspect of ontologymediation. In a nutshell, given two ontologies, GLUE finds for each concept in one ontology the most similar concept in the other ontology. The method used exploits several matchers and probabilistic definitions of several similarity measures. Like our solution, the mediation solution requires the intervention of end-users in the beginning of the process: selecting the data to train the matchers in GLUE and selecting instances of the ABoxes to design the input matrix for the Galois connection algorithm in our solution. But we consider that this task in GLUE is more demanding than in our solution as both positive and negative information are generally required to train machine learning effeciently.

Considering works involving FCA methods and DLs, it is interesting to study [Baader et al. 2007]. In this paper, the authors are concerned with the completeness quality dimension of TBoxes, i.e. they propose techniques to enable ontology engineers in checking if all the relevant concepts of an application domain are present in a TBox. Like our approach, one of their concern is to minimize interactions with domain experts. Hence FCA techniques are being used to withdraw trivial questions that may be asked to experts in case of incomplete TBoxes. The approach we presented in this paper is more concern with the generation and optimization of mediated ontology. And we can consider that our approach is more involved in the soundness quality dimension and tackles the issue of generating different forms of merged ontology.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an approach to mediate, i.e. alignment and merging, ontologies based on the methods of FCA. Our main contribution include (i) the possibility to create concepts not originally in the source ontologies but which emerge from their merging, (ii) the ability to provide labels for these new concepts, based on the labels of the implied source concepts, (iii) the optimization of the mediated ontology by eliminating redundant and non-pertinent concepts and finally (iv) to emphasize that several mediated ontologies can be defined from our solution and that some measures can help end-users to select the most appropriate one in a given context.

We tested our solution in the domain of medical informatics with drug related ontologies. This context was particularly adapted to our approach since some of the objects of the ABoxes are drug products of the French market distinguished by a common identifying solution. The method showed particularly useful for a medical informatics project we are working on with health care professionals. But a deep understanding of the concept of ontology engineering is required to select the most adapted form of merged ontology for an application. So currently, it is not possible to let the team of health experts interact directly with the system without an ontology expert. We are now planning to test our solution in the geography domain with ontologies related to landscape description and analysis.

We would also like to use recent studies and results in automatic labeling of anonymous data. For example, the work presented in [da Silva et al. 2007] uses a probabilistic model for estimating the affinity between attributes and candidate labels. We believe that such solutions would enable to generate labels using more relevant words than those used in the source ontologies.

Future work on this system are related to extracting automatically a valuable and minimal set of instances from ABoxes for the Galois connection matrix and providing axioms to the merged ontology according to the axioms retrieved from the source ontologies. This approach will enable us to deal with expressive DLs.

REFERENCES

ACKOFF, R. 1989. From data to wisdom. *Journal of Applied Systems Analysis* 16:3–9.

BAADER, F., D. CALVANESE, D. L. MCGUINNESS, D. NARDI, AND P. F. PATEL-SCHNEIDER. Eds. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press.

BAADER, F., B. GANTER, B. SERTKAYA, AND U. SATTLER. 2007. Completing description logic knowledge bases using formal concept analysis. In *IJCAI* (2007-03-05), M. M. Veloso, Ed. 230–235.

BECKER, P. AND J. H. CORREIA. 2004. The toscanaj suite for implementing conceptual information systems. In *Formal Concept Analysis State of the Art, Berlin Heidelberg.* Springer.

BERNERS-LEE, T., J. HENDLER, AND O. LASSILA. 2001. The semantic web: Scientific american. *Scientific American.*

DA SILVA, A. S., D. BARBOSA, J. M. B. CAVALCANTI, AND M. A. S. SEVALHO. 2007. Labeling data extracted from the web. In *OTM Conferences (1).* 1099–1116.

DAVEY, B. A. AND H. A. PRIESTLEY. 2002. *Introduction to Lattices and Order.* Cambridge University Press.

DOAN, A., J. MADHAVAN, P. DOMINGOS, AND A. HALEVY. 2002. Learning to map between ontologies on the semantic web. In *WWW '02: Proceedings of the 11th international conference on World Wide Web.* ACM, New York, NY, USA, 662–673.

DONG, X., A. HALEVY, AND J. MADHAVAN. 2005. Reference reconciliation in complex information spaces. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data.* ACM, New York, NY, USA, 85–96.

DOU, D., D. MCDERMOTT, AND P. QI. 2002. Ontology translation by ontology merging and automated reasoning. In *Proc. EKAW Workshop on Ontologies for Multi-Agent Systems.* 3–18.

EHRIG, M. 2006. *Ontology Alignment: Bridging the Semantic Gap (Semantic Web and Beyond).* Springer-Verlag New York, Inc., Secaucus, NJ, USA.

ELFEKY, M. G., A. K. ELMAGARMID, AND V. S. VERYKIOS. 2002. Tailor: A record linkage tool box. In *ICDE.* IEEE Computer Society, 17–28.

EUZENAT, J. AND P. SHVAIKO. 2007. *Ontology matching.* Springer-Verlag, Heidelberg (DE).

FELLEGI, I. P. AND A. B. SUNTER. 1969. A theory for record linkage. *Journal of the American Statistical Association 64,* 328:1183–1210.

GANTER, B. AND R. WILLE. 1999. *Formal Concept Analysis – Mathematical Foundations.* Springer.

HUSTADT, U., B. MOTIK, AND U. SATTLER. 2004. Reducing shiq-description logic to disjunctive datalog programs. In *KR,* D. Dubois, C. A. Welty, and M.-A. Williams, Eds. AAAI Press, 152–162.

KALFOGLOU, Y. AND M. SCHORLEMMER. 2003. Ontology mapping: the state of the art. *Knowl. Eng. Rev. 18,* 1:1–31.

KANELLAKIS, P. C. 1990. Elements of relational database theory. 1073–1156.

MCGUINNESS, D. L. 2003. *Ontologies Come of Age.* MIT Press.

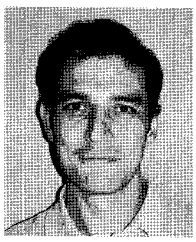
MOTIK, B., I. HORROCKS, AND U. SATTLER. 2007. Bridging the gap between owl and relational databases. In *WWW.* 807–816.

NOY, N. F. AND M. A. MUSEN. 2000. Prompt: Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence.* AAAI Press/The MIT Press, 450–455.

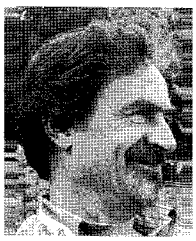
PENG LIM, E., S. PRABHAKAR, J. SRIVASTAVA, AND J. RICHARDSON. 1993. Entity identification in database integration. In *Proceedings Ninth International Conference on Data Engineering.* IEEE Computer Society Press, 294–301.

POGGI, A., D. LEMBO, D. CALVANESE, G. D. GIACOMO, M. LENZERINI, AND R. ROSATI. 2008. Linking data to ontologies. *J. Data Semantics 10,* 133–173.

STUMME, G. AND E. MAEDCHE. 2001. Fca-merge: Bottom-up merging of ontologies. In *IJCAI.* 225–230.



Olivier Curé is an assistant professor in computer science at the Université Paris-Est in France. He obtained his Ph.D. in artificial intelligence at the Université de Paris V, France.
He has published 4 book chapters, 4 journal papers and more than 35 papers in international, peer-reviewed conferences on databases, semantic web and ontologies. He has organized a one day workshop, named ADI (Ambient Data Integration) at OTM'08 and will organize a second edition at OTM'09. He has received grants, together with Pr. Stefan Jablonski, to establish a research cooperation with the University of Bayreuth, Germany. He is currently heading the Terre Digitale (Digital Earth) research team at the Universite of Paris-Est which is regrouping researchers in computer science and geographic information.



Robert Jeansoulin is a senior scientist in the computer science sector at CNRS. Since july 2008, he is an Attaché for science & technology (IT), Embassy of France, Washington.
Before that, his last position was at Université Paris-Est (Marne-la-Vallée), teaching Data Bases, and pursuing research in Artificial Intelligence.
He has published 4 books, 18 journal articles, and more than 60 papers in international, peer-reviewed conferences on image processing, spatial data bases, and artificial intelligence. He has supervised 20 PhD candidates, several of whom are now professors in different universities, or researchers in laboratories in France, Canada, Ireland, Tunisia.
He has received awards for papers (e.g. from the ASPRS in 2006) and software releases (from IGN in 1992, from Intergraph in 2004).