Ontology refinement through role assertion analysis: example in pharmacogenomics

Adrien Coulet*, Malika Smaïl-Tabbone, Amedeo Napoli, and Marie-Dominique Devignes

LORIA - UMR 7503, Nancy-University, CNRS, INRIA,
F 54506 Vandoeuvre-lès-Nancy, France
{coulet,malika,napoli,devignes}@loria.fr

Abstract. We propose an approach for extending domain knowledge represented in DL ontology by using knowledge extraction methods on ontology assertions. Concept and role assertions are extracted from the ontology in the form of assertion graphs, which are used to generate a formal context manipulated by Formal Concept Analysis methods. The resulting expressions are then represented as DL concepts and roles that can be inserted into the initial ontology after validation by the analyst. We show, through a real-world example, how this approach has been successfully used for discovering new knowledge units in a pharmacogenomics ontology.

Key words: DL ontology, FCA, Ontology refinement, Pharmacogenomics

1 Introduction

At present, many resources in a given domain, e.g. life science, are available. These resources have many forms: databases, thesauri, ontologies, documents, etc. One objective of Knowledge Discovery in Databases (KDD) methods is to extract reusable and significant knowledge units from such resources. Advances in Semantic Web technologies promotes the formalisation and management of such knowledge units within Description Logics (DL) ontologies. A great challenge is to take advantage of these formal ontologies for guiding knowledge discovery [1]. In this paper, we present an original KDD process carried out in the context of a DL ontology. The objective is to extract knowledge units from assertions (involving individuals) lying in the ontology. We propose to apply Formal Concept Analysis methods for extracting regularities from ontology assertions that will be used to refine the initial ontology. To achieve this task, assertion graphs, connecting ontology individuals, are used as a basis for generating a formal context, then manipulated by FCA methods.

Contrasting current methods of ontology refinement based on Natural Language Processing using text corpus as knowledge resources [2], this work proposes the analysis of existing assertions for refining an ontology. In [3], a study is presented on the use of FCA for computing the hierarchical structure of an ontology. More recently, in [4], a complement study shows how to complete the terminological as well as the assertional

* corresponding author

part of ontologies. In [5] FCA and Description Logics are combined with different objectives for enriching a concept lattice (within *augmented context*) on the basis of DL knowledge. In this paper, we use concept lattice to acquire new knowledge in a DL ontology. We firstly present the motivation of our approach, secondly introduce notions of DL and FCA needed for understanding the paper. Third, we detail our orignial approach that we call Role Assertion Analysis (RAA). Then we finally illustrate it with a life science application.

2 Motivation

Life science processes are addressed in many sophisticated methods that produce large volumes of complex and highly interrelated data. It is consequently critical in this domain to take into account amounts of available data and knowledge. A typical example is pharmacogenomics that studies the implication of inter-individual genomic variations in drug response. Data and knowledge relative to genomic variations, drugs, genotype-phenotype interactions, and clinical trials have to be exploited by expert analysts to enlighten on hidden relationships between drug treatments, genomic variations, and phenotype traits.

A pharmacogenomic ontology, called IDANAT2 ONTOLOGY, was produced in the present study for illustrating our ontology refinement approach. The IDANAT2 ONTOLOGY is inspired from SO-PHARM, which was previously described in [7] and is available on the OBO-Foundry web site [8]. The IDANAT2 ONTOLOGY contains twenty six concepts and twenty roles about pharmacogenomics: e.g. concepts that represent drug treatment, phenotype, and genotype of patients. The ontology is instantiated on the basis of a real pharmacogenomic clinical trial: IDANAT2, which is performed to study involvement of variations in the *NAT2* gene in responses to tuberculosis treatment with isoniazid (inh) [9]. A total of 12 patients, with their treatments, their genotypes, and phenotypes have been used for instantiating the IDANAT2 ONTOLOGY.

Our hypothesis is that novel knowledge units relevant in the field of pharmacogenomics are hidden inside assertions. Expected ontology refinements are followings: characterisation of the patient panel (e.g. all patients are smokers), identification of subgroups reacting differently (e.g. patients with slow/rapid drug metabolism), and identification of relations between treatment, genotype and phenotype factors (e.g. high dose of inh, version 5* of the gene *NAT2*, and an adverse response).

3 DL ontology and Formal Concept Analysis

3.1 DL ontology

A DL ontology is a representation of a domain knowledge expressed in DL formalism [10]. A DL ontology usually consists of two parts: the terminological part or TBox $\mathcal{T}$, and the assertional part or ABox $\mathcal{A}$. TBox defines concepts in relation with other concepts according to hierarchical and non-hierarchical relationships. The semantics of concept descriptions is defined in terms of an *interpretation* $\mathcal{I}$ that consists of a non-empty set $\Delta^{\mathcal{I}}$ and an interpretation function, which assigns to every concept $\mathtt{C}$ a

set $\mathtt{C}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to every role R a binary relation between a domain and a co-domain $\mathtt{R}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. A role R can have an inverse role $\mathtt{R}^{-}$ whose interpretation is

$$(\mathtt{R}^{-})^{\mathcal{I}} = \{(b,a) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} | \, (a,b) \in \mathtt{R}^{\mathcal{I}}\}$$

The ABox includes individuals and their relationships in the form of *concept* and *role assertions*. Such assertions are formula that describe the instanciation of concept and roles with individuals. An example of concept assertion is: "`Patient` (*pa*01)" meaning that *pa*01 is an instance of the concept `Patient`. An example of role assertion is: "`hasClinicalItem` (*pa*01, *tuberculosis*)" meaning that the individual *pa*01 is related to the individual *tuberculosis* through the role `hasClinicalItem` .

The term *DL ontology* used in this paper, refers conjointly to TBox and ABox also called *DL knowledge base*.

In this paper two specific constructors of DL are used. First, the "fills" constructor, denoted by $\mathtt{R} : b$, that is a concept constructor associating individual names to the co-domain of a role R . Its semantics is defined as

$$(\mathtt{R} : b)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} | \, (a, b^{\mathcal{I}}) \in \mathtt{R}^{\mathcal{I}}\} \tag{1}$$

and b is said to be the "filler" of R. $\mathtt{R} : b$ is equivalent to the more usual "`hasValue`" notation $\exists \, \mathtt{R}.\{b\}$. Second, the role composition constructor, that is a role constructor, denoted by $\mathtt{R} \circ \mathtt{S}$, for two roles R and S is interpreted as

$$(\mathtt{R} \circ \mathtt{S})^{\mathcal{I}} = \{(a,c) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} | \, \exists b.(a,b) \in \mathtt{R}^{\mathcal{I}} \wedge (b,c) \in \mathtt{S}^{\mathcal{I}}\}. \tag{2}$$

The combination of the two constructors produces an expression such as $\mathtt{R} \circ \mathtt{S} : b$, which is equivalent to $\exists \, \mathtt{R}.(\exists \, \mathtt{S}.\{b\})$.

3.2 Formal Concept Analysis

Formal Concept Analysis (FCA) is the process of abstracting conceptual descriptions from a set of objects described by attributes [11]. Formally, FCA studies a *formal context* $\mathcal{K}$ that associates a set of objects $\mathcal{G}$ to a set of attributes $\mathcal{M}$ through an incidence relation $\mathcal{I}_{\mathcal{R}}$ in $\mathcal{G} \times \mathcal{M}$. An example of formal context is depicted in Table 1, where $\mathcal{G}$ is a set of patients and $\mathcal{M}$ a set of patient attributes. A *formal concept*, issued from a formal context $\mathcal{K} = (\mathcal{G}, \mathcal{M}, \mathcal{I}_{\mathcal{R}})$, is defined as a pair (A, B) where A is a subset of objects, the *extent* of the concept, and B is a subset of attributes, the *intent* of the concept. The definition of the concept (A, B) is based on a Galois connection defined by the dual $'$ application that transforms a set of objects A and a set of attributes B as follows

$$A' := \{m \in \mathcal{M} | \, \forall a \in A : (a,m) \in \mathcal{I}_{\mathcal{R}}\},$$

$$B' := \{g \in \mathcal{G} | \, \forall b \in B : (g,b) \in \mathcal{I}_{\mathcal{R}}\}.$$

A formal concept (A, B) verifies $A = B'$ and dually $B = A'$. Formal concepts can be hierarchically ordered by inclusion of their extents. This order induces a lattice, called the *concept lattice* of $\mathcal{K}$. Such a lattice is always complete meaning that for any pair of

concepts there exists a unique smallest superconcept and a unique largest subconcept. An example of concept lattice is given in Figure 5.

Thanks to its mathematical properties, a concept lattice can be used as an intermediate support structure to mine multiple data dependencies, such as Association Rules $\mathcal{AR}$, that hold in a context [12]. $\mathcal{AR}$ are probabilistic data dependencies having the form $B_1 \Rightarrow B_2$. B_1 (the *antecedent*) and B_2 (the *consequent*) are sets of attributes and $B_1 \Rightarrow B_2$ expresses that the presence of attributes in B_1 implies the presence of attributes in B_2 with a specified *support* and *confidence*:

$$support(B_1 \Rightarrow B_2) = \frac{|(B_1 \cup B_2)'|}{|\mathcal{G}|}$$

$$confidence(B_1 \Rightarrow B_2) = \frac{|(B_1 \cup B_2)'|}{|B_1'|}$$

$\mathcal{AR}$ are usually computed from Frequent Itemsets, i.e. sets of attributes with a support greater than a *minimum support*. Since the number of $\mathcal{AR}$ extracted from a formal context can be huge , reduced sets of $\mathcal{AR}$ need to be identified [13] [14]. In this work, we used the set of *Reduced Minimal Non-Redundant Rules* $\mathcal{RMNR}$ [15], which is the transitive reduction of *Minimal Non-Redundant Rules* $\mathcal{MNR}$ defined by Kryszkiewicz in [14]. $\mathcal{RMNR}$ represents a reduced set of rules with a minimal antecedent and a maximum consequent, from which all other rules can be derived. The calculation of $\mathcal{RMNR}$ used in this work takes benefit of the intermediate structure provided by a formal lattice. This calculation is based on the Next Closure Algorithm of B. Ganter [13].

4 Role Assertion Analysis

Role Assertion Analysis (RAA) is a semi-automatic process. It takes as input a DL ontology $\mathcal{O} = (\mathcal{T}, \mathcal{A})$, a DL concept description C_0, and a parameter named maximum depth d_{max}, and returns as output a refined version of the original ontology $\mathcal{O}$ with new concept descriptions, new roles, and new role assertions.

Figure 1 depicts the characteristics of this approach that could be divided in four main steps:

(a) extraction of the set of assertion graphs associated with a specified concept C_0,
(b) transformation of assertion graphs into a formal context,
(c) analysis of the formal context with FCA, and
(d) interpretation of FCA results in terms of new DL concepts and roles.

These four steps could be compared to extraction, transformation, mining, and interpretation steps, classically described in the KDD process. This is detailed in next subsections.

4.1 From ontology to assertion graphs

Firstly, the analyst has to define a concept C_0 within the ontology $\mathcal{O}$ from which the set of instances $\mathcal{A}_0$ will be considered for RAA. In practice, the description of C_0 is not

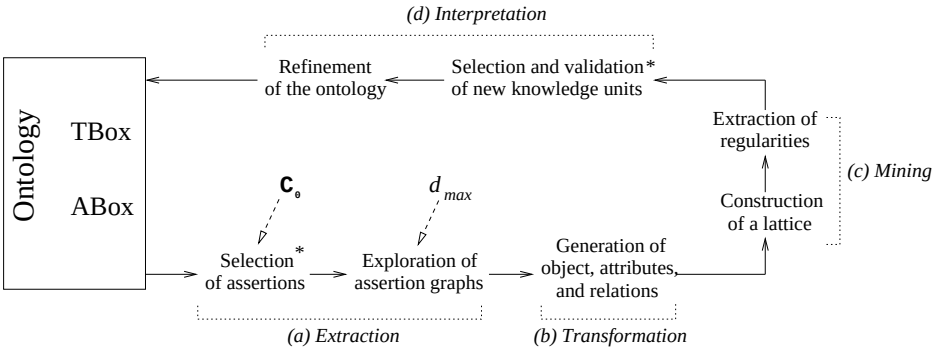

Fig. 1. Overview of the Role Assertion Analysis (RAA) approach. Action tagged with * denotes interactive steps.

constrained. It can be a concept that is explicitly defined in $\mathcal{O}$ as well as an undefined concept, described with DL constructors, and concepts from $\mathcal{T}$. $\mathcal{A}_0$ is defined as the set of individuals a that are instances of $\mathtt{C}_0$ thus satisfying $\mathcal{A}_0 = \{a \mid \mathtt{C}_0(a)\}$.This manual step is cucial since it enables the analyst to focus on a subset of individuals relevant to characterise. Section 5 give an example of a concept $\mathtt{C}_0$.

We define the *assertion graph* $G_a = (V, E)$ as a rooted oriented cyclic graph with a as *root vertex*, concept assertions as *vertices* V, and role assertion as *edges* E. The *assertion graph* $G_a = (V, E)$ of each individual a from $\mathcal{A}_0$, is then explored in order to compute every possible path that relates, through role assertions, individual a to other individuals in $\mathcal{O}$. This graph is explored by a depth-first search algorithm constrained by one parameter, the maximum depth d_{max} given as an input by the analyst (e.g. $d_{max} = 3$), and two restrictions: (1) it is not allowed to pass more than once by a same vertex, (2) it is not allowed to traverse an edge (that corresponds to $\mathtt{R}$) and then the edge associated to the inverse relation (that corresponds to $\mathtt{R}^-$). Parameter d_{max} limits the depth progression of the algorithm to a maximum number of edges within a single path. Constraint (1) guarantees the absence of cycle in generated paths, constraint (2) is a practical choice that reduces the size of the generated formal context. It can be demonstrated that under this conditions, the assertion graph is finally totally (but not minimally) covered by paths at a maximal d_{max} depth. An example of assertion graph and all its derived paths is displayed in Figure 2.

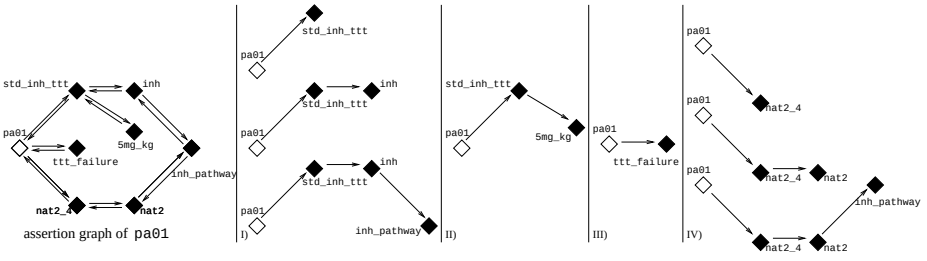

Fig. 2. Schematization of the assertion graph of pa01 and of each generated path with $d_{max} = 3$

At the end of graph exploration, each individual a is associated with a set of paths and sub-paths that can be described as DL concepts. For example, a path that follows

successively edges corresponding to roles R and S and ends on the vertex corresponding to an individual b that instantiate the concept C_b is a representation, in the graph, of the concept $R \circ S : b$. This is interpreted using constructors (1) and (2) defined in subsection 3.1 as

$$(R \circ S : b)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid (a, b^{\mathcal{I}}) \in (R \circ S)^{\mathcal{I}}\}. \quad (3)$$

4.2 From assertion graphs to formal context

The task consists here in transforming the set of assertion graphs $G_a = (V, E)$ into a formal context $\mathcal{K} = (\mathcal{G}, \mathcal{M}, \mathcal{I}_{\mathcal{R}})$ that can be subsequently analysed with FCA methods. Obviously enough, the set of object $\mathcal{G}$ corresponds to the set of individuals $\mathcal{A}_0$. Mapping the assertion graph G_a to a set of attributes $\mathcal{M}$ according to the incidence relation $\mathcal{I}_{\mathcal{R}}$ is more complex. For such a given graph, each computed path or sub-path (described as a DL concept) is translated into strings that label attributes to add to $\mathcal{M}$. For example the expression (3) above is used to create at least two attributes m_x, m_C in $\mathcal{M}$ identified by the following strings

$m_x \coloneqq$ R_o_S:b means that a is related to b by an assertion of $R \circ S$,
$m_C \coloneqq$ R_o_S:C_b means that a is related to an assertion of C_b by $R \circ S$.

The attribute m_C is generated in the hope that more objects will display attribute m_C than m_x and will then contribute to produce more regularities in the context. Since b could instantiate several concepts of $\mathcal{O}$, one path could generate several attributes of the form of m_C. The same path is also used to fill the incidence relation $\mathcal{I}_{\mathcal{R}}$ of the context $\mathcal{K}$. If a path from a leads to the creation of attributes m_x and m_C, then the incidences $a\mathcal{I}m_x$ and $a\mathcal{I}m_C$ are filled in $\mathcal{I}_{\mathcal{R}}$. The formal context presented in Table 1 results from this first step.

4.3 From formal context to Reduced Minimal Non-Redundant Rules

The third step of the proposed approach consists in the extraction of regularities from the formal context. Regularities are typically deterministic or probabilistic dependences between attributes. The first task for analysing such regularities in the context is to compute the concept lattice (see Figure 4 for an example). A classical algorithm to construct a lattice is the Next Closure Algorithm [13]. This algorithm has been recently refined and compared to other existing ones by Kuznetsov and Obiedkov in [16]. Once the concept lattice is constructed, its mathematical structure can be analysed by different methods.This study is based on $\mathcal{RMNR}$ (see subsection 3.2), because $\mathcal{RMNR}$ yields the smallest set of rules from which all valid rules can be derived. $\mathcal{RMNR}$ is smaller than closed association rules and minimal non-redundant association rules. Of course, depending on the interest of the analyst, other regularities, such as rare association rules, could be extracted from the context [17].

4.4 From Reduced Minimal Non-Redundant Rules to new DL concepts and roles

The last step involves again the analyst since she has to select a subset of relevant attributes from $\mathcal{RMNR}$ for defining new DL concepts and roles. This last step presents similarities with emerging work on Relational Concept Analysis (RCA) [18].

New DL concepts Selected attributes within a rule are translated back and expressed as a DL concept. Formally, if m_b,$m_d \subseteq \mathcal{M}$ are selected within a rule, they are expressed in DL as:

$$m_b \text{ with the label } \texttt{R\_o\_S:b} \text{ is expressed as } \texttt{R} \circ \texttt{S} : b,$$
$$m_d \text{ with the label } \texttt{T\_o\_U:d} \text{ is expressed as } \texttt{T} \circ \texttt{U} : d,$$

and associated as follow in $\texttt{C}_{\texttt{new}} \coloneqq \texttt{R} \circ \texttt{S} : b \sqcap \texttt{T} \circ \texttt{U} : d$. This task is not limited to association of two attributes or to compositions of two roles. Attributes of the form of $m_{\texttt{C}}$ are similarly translated back in DL but without using the "fills" constructor. For example, attribute $\texttt{R\_o\_S:C}_\texttt{b}$ is translated back in $\exists\, \texttt{R} \circ \texttt{S}.\texttt{C}_b$.

Once a new concept $\texttt{C}_{\texttt{new}}$ is defined, the last step is its insertion into the original ontology. To achieve this task, the most specific concept $\texttt{C}_{\texttt{subs}}$ from $\mathcal{T}$ that satisfies $\vDash_{\mathcal{O}} \texttt{C}_{\texttt{new}} \sqsubseteq \texttt{C}_{\texttt{subs}}$ is computed. Then, depending on analyst validation, $\texttt{C}_{\texttt{new}}$ becomes a new concept in $\mathcal{O}$ such as $\texttt{C}_{\texttt{new}} \sqsubseteq \texttt{C}_{\texttt{subs}}$, or alternatively $\texttt{C}_{\texttt{new}}$ description completes $\texttt{C}_{\texttt{subs}}$ description: $\texttt{C}_{\texttt{subs}} \equiv \texttt{C}_{\texttt{new}}$ (that could also be noted $\texttt{C}_{\texttt{subs}} \equiv \texttt{C}_{\texttt{subs}} \sqcap \texttt{C}_{\texttt{new}}$) . The latter occurs when the analyst observes that $\texttt{C}_{\texttt{new}}$ definition is more precise in describing what is supposed to be represented by $\texttt{C}_{\texttt{subs}}$. Section 5 illustrates this interpretation step. We propose an algorithm that formalises this task in [19].

New roles and/or role assertions Primarily selected attributes also lead to creating new roles and/or new role assertions. Selected attributes are expressed as DL concept descriptions as for the creation of new DL concepts, but are not associated and are stored independently. The analyst is then proposed to create new roles $\texttt{R}_{new}$ between each pair of "fillers"(see subsection 3.1) of new DL concepts, e.g. b and d. It is interesting to notice that "fillers" correspond to last vertices of paths explored in the assertion graph, and to individuals that are linked to individuals from $\mathcal{A}_0$. A creation of a new role is proposed if no role exists in the ontology that can be asserted to link "fillers" directly. In this case, the new role $\texttt{R}_{new}$ is directly instanciated with corresponding "fillers", e.g. $\texttt{R}_{new}(b, d)$. In the alternative case where role already exists, the analyst chooses either to instanciate an existing role if one is relevant or to create and instanciate a new one. [19] gives an algorithm that details operations of this task.

5 Role Assertion Analysis in pharmacogenomics

RAA has been applied to a real-world ontology in pharmacogenomics called IDANAT2 ONTOLOGY (introduced in section 2) with the objective of discovering new knowledge units concerning the pharmacogenomics of the gene *NAT2*. A view of some concepts, concept and role assertions of IDANAT2 ONTOLOGY is given in Figure 3.

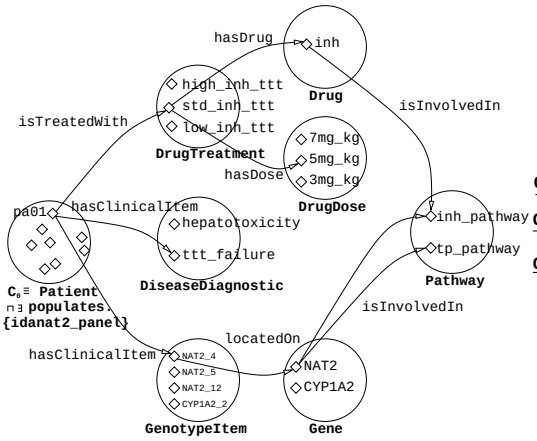

Fig. 3. A view of IDANAT2 ONTOLOGY

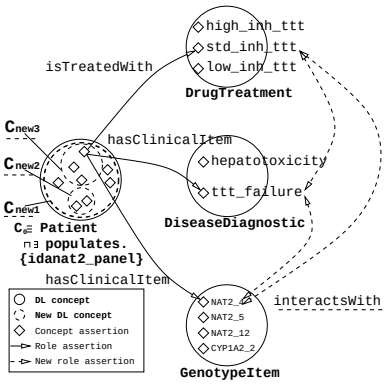

Fig. 4. A view of the refinement

The analyst initially selects the group of all patients included in the clinical study IDANAT2. This subgroup of instances of the concept Patient is defined in C_0 as

$$\mathsf{C}_0 \equiv \texttt{Patient} \sqcap \exists_{\texttt{populates}}.\{idanat2\_panel\},$$

C_0 is instanciated by twelve individuals that corresponds to the twelve patients involved in the study. Assuming a maximum depth $d_{max} = 3$, Figure 2 depicts the assertion graph relative to the assertion $\mathsf{C}_0(pa01)$. The figure distinguishes eight different paths explored in the graph that reflect all relations that $pa01$ has with other individuals in IDANAT2 ONTOLOGY. The resulting context $\mathcal{K}_{IDANAT2}$ (as explained in Section 4.2) is displayed in Table 1. The set of objects $\mathcal{G}$ contains all patients of the clinical trial, and the set $\mathcal{M}$ contains twenty-two attributes produced from paths covering all assertion graphs. For example, the path located in Figure 2 III) is translated into the following attributes

$$m_x := \texttt{hasClinicalItem:tuberculosis}$$
$$m_{\mathsf{C}} := \texttt{hasClinicalItem:DiseaseDiagnostic}.$$

The concept lattice constructed for the context $\mathcal{K}_{IDANAT2}$ is presented in Figure 5. Each node stands for a concept harbouring *owned attributes* (displayed on the top of the node), which are attributes that distinguish the concept from upper ones, and *inherited attributes*, which are inherited from upper concepts. For example, the first concept on the left is interpreted as the subgroup of patients $pa02$, $pa03$, and $pa01$ sharing the local owned attribute (hasClinicalItem:ttt_failure) that does not appear in any ascendant concept, and sharing inherited attributes from upper concepts, such as hasClinicalItem:NAT2_4.

$\mathcal{RMNR}$ are computed with a *minimum support* of 0.25 and a *minimum confidence* of 0.8. Consequently, the concept lattice leads to identify six distinct $\mathcal{RMNR}$. According to the analyst, three of these rules describe relationships which are already known, already in the ontology, and therefore out of the scope of the study. The three other rules were treated according to the process described in subsection 4.4. Table 2 describes how they are, first, translated back in three DL concepts $\mathsf{C}_{\texttt{new1}}, \mathsf{C}_{\texttt{new2}}, \mathsf{C}_{\texttt{new3}}$, and second, articulated with concepts of the original ontology. From the analyst point of view, $\mathsf{C}_{\texttt{new1}}$

Table 1. Formal context $\mathcal{K}_{IDANAT2}$ constructed on the basis of the DL ontology IDANAT2 ONTOLOGY

$\mathcal{G}$ \ $\mathcal{M}$	isTreatedWith:DrugTreatment	isTreatedWith:std_inh_ttt	isTreatedWith_o_hasDrug:Drug	isTreatedWith_o_hasDrug:inh	isTreatedWith_o_hasDrug_o_involvedIn: Pathway	isTreatedWith_o_hasDrug_o_involvedIn: inh_pathway	isTreatedWith_o_hasDose:DrugDose	isTreatedWith_o_hasDose:5mg_kg	hasClinicalItem:DiseaseDiagnostic	hasClinicalItem:ttt_failure	hasClinicalItem:hepatotoxicity	hasClinicalItem:GenotypeItem	hasClinicalItem:NAT2_4	hasClinicalItem:NAT2_12	hasClinicalItem:NAT2_5	hasClinicalItem:CYP1A2_2	hasClinicalItem_o_locatedOn:Gene	hasClinicalItem_o_locatedOn:NAT2	hasClinicalItem_o_locatedOn:CYP1A2	hasClinicalItem_o_locatedOn_o_involvedIn: Pathway	hasClinicalItem_o_locatedOn_o_involvedIn: inh_pathway	hasClinicalItem_o_locatedOn_o_involvedIn: tp_pathway
pa01	×	×	×	×	×	×	×	×	×	×		×	×			×	×	×	×	×	×	×
pa02	×	×	×	×	×	×	×	×	×	×		×	×				×	×		×	×	
pa03	×	×	×	×	×	×	×	×	×	×		×	×				×	×		×	×	
pa04	×	×	×	×	×	×	×	×	×			×	×	×			×	×		×	×	
pa05	×	×	×	×	×	×	×	×	×			×	×	×			×	×		×	×	
pa06	×	×	×	×	×	×	×	×	×			×	×	×			×	×		×	×	
pa07	×	×	×	×	×	×	×	×	×			×		×		×	×	×	×	×	×	×
pa08	×	×	×	×	×	×	×	×	×			×		×	×		×	×		×	×	
pa09	×	×	×	×	×	×	×	×	×			×		×	×		×	×		×	×	
pa10	×	×	×	×	×	×	×	×	×		×	×			×		×	×		×	×	
pa11	×	×	×	×	×	×	×	×	×		×	×			×		×	×		×	×	
pa12	×	×	×	×	×	×	×	×	×		×	×			×	×	×	×	×	×	×	×

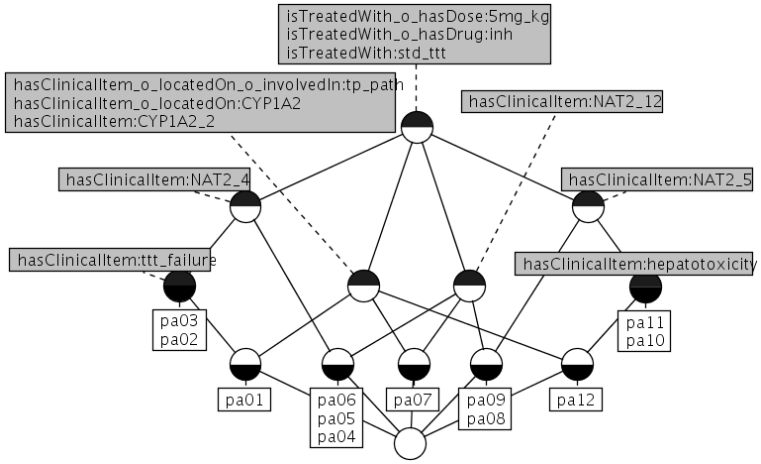

Fig. 5. Concept lattice for the context $\mathcal{K}_{IDANAT2}$ with reduced labelling. The figure displays only tree of fourteen attributes of the top node.

enables to characterise patients treated with inh (*inh*) with a daily dose of 5mg per kg of body weight (*5mg_kg*). $\mathsf{C}_{\mathtt{new2}}$ and $\mathsf{C}_{\mathtt{new3}}$ describe two relevant subgroups within the

panel. $\mathsf{C}_{\texttt{new2}}$ stands for *slow acetylators* that metabolize inh slowly thus accumulating toxic metabolites in their liver (*hepatotoxicity*). Inversely, $\mathsf{C}_{\texttt{new3}}$ stands for *rapid acetylators* that metabolize the drug so quickly that it has no effect (*ttt_failure*). Each subgroup is associated with a different version of *NAT2* gene (*NAT2_5* and *NAT2_4*).

Table 3 describes pairs of "fillers" extracted from rules and validated or not (denoted with $\emptyset$) to assert roles in the ontology. For example, the rule $\mathcal{AR}_3$ enables to identify three pairs of "fillers". No role assertion corresponds to these pairs, but the existing role `interactsWith` can be asserted to link pair members. The analyst proposes to assert this role with each pair. Resulting role assertions between treatment, genotype and phenotype are typical relevant knowledge units in pharmacogenomics.

Resulting new concepts and new roles assertion (those generated from $\mathcal{AR}_3$) are schematized in Figure 4.

Table 2. From $\mathcal{RMNR}$ to new concept in the TBox $\mathcal{T}$

Rule	*DL concept description*	*Refinement of* $\mathcal{T}$
$\mathcal{AR}_1 \rightarrow \mathsf{C}_{\texttt{new1}} \equiv$	$\exists$ isTreatedWith .{std_inh_ttt}	$\rightarrow \mathsf{C}_{\texttt{new1}} \sqsubseteq \mathsf{C}_{\texttt{subs}}$
	$\sqcap\exists$ isTreatedWith.($\exists$ hasDrug.{inh})	
	$\sqcap\exists$ isTreatedWith.($\exists$ hasDose.{5mg_kg})	
$\mathcal{AR}_2 \rightarrow \mathsf{C}_{\texttt{new2}} \equiv$	$\exists$ isTreatedWith .{std_inh_ttt}	$\rightarrow \mathsf{C}_{\texttt{subs}} \equiv \mathsf{C}_{\texttt{subs}} \sqcap \mathsf{C}_{\texttt{new2}}$
	$\sqcap\exists$ hasClinicalItem .{hepatotoxicity})	
	$\sqcap\exists$ hasClinicalItem .{NAT2_5})	
$\mathcal{AR}_3 \rightarrow \mathsf{C}_{\texttt{new3}} \equiv$	$\sqcap\exists$ hasClinicalItem .{ttt_failure})	$\rightarrow \mathsf{C}_{\texttt{new3}} \sqsubseteq \mathsf{C}_{\texttt{subs}}$
	$\sqcap\exists$ hasClinicalItem .{NAT2_4})	

Table 3. From $\mathcal{RMNR}$ to new role assertions in the ABox $\mathcal{A}$

Rule	*Pair of "fillers"*	*Refinement of* $\mathcal{A}$
$\mathcal{AR}_1 \rightarrow$	(std_inh_ttt,inh),	$\rightarrow \emptyset$,
	(std_inh_ttt,5mg_kg),	$\emptyset$,
	(inh,5mg_kg)	$\emptyset$
$\mathcal{AR}_2 \rightarrow$	(std_inh_ttt,hepatotoxicity)	$\rightarrow \emptyset$,
	(std_inh_ttt,NAT2_5)	interactsWith(std_inh_ttt,NAT2_5),
	(hepatotoxicity,NAT2_5)	interactsWith(hepatotoxicity,NAT2_5)
$\mathcal{AR}_3 \rightarrow$	(std_inh_ttt,ttt_failure)	$\rightarrow$ interactsWith(std_inh_ttt,ttt_failure),
	(std_inh_ttt,NAT2_4)	interactsWith(std_inh_ttt,NAT2_4),
	(ttt_failure,NAT2_4)	interactsWith(ttt_failure,NAT2_4)

6 Conclusion

The novelty of the RAA process described in this paper relies in the exploitation of role assertions as a basis for mining ontology instances. We have illustrated through a

real-world example how this approach can succeed in discovering several knowledge units that were implicitly embedded in ontology instances. Implementation of RAA is underway in the frame of a plug-in for the ontology editor Protégé 4. As for any KDD process RAA can be run in an iterative manner. First experimentations show that relevant refinements of ontology occur gradually during successive iterations of RAA process.

References

1. Lieber, J., Napoli, A., Szathmary, L., Toussaint, Y.: First Elements on Knowledge Discovery Guided by Domain Knowledge (KDDK). In: Proc. of CLA06. LNAI 4923 (2008) 22–41
2. Buitelaar, P., Cimiano, P., Magnini, B.: Ontology Learning from Text: Methods, Evaluation and Applications Frontiers. IOS Press (2005)
3. Baader, F., Sertkaya, B.: Applying formal concept analysis to description logics. In: Proc. of ICFCA'04. LNCS 2961 (2004) 261–286
4. Baader, F., Ganter, B., Sertkaya, B., Sattler, U.: Completing description logic knowledge bases using formal concept analysis. In: Proc. of IJCAI'07. (2007) 230–235
5. Ferré, S., Ridoux, O., Sigonneau, B.: Arbitrary relations in formal concept analysis and logical information systems. In: Proc. of ICCS'05. LNCS 3596 (2005) 166–180
6. Hacene, M.R., Huchard, M., Napoli, A., Valtchev, P.: A proposal for combining formal concept analysis and description logics for mining relational data. In: Proc. of ICFCA'07. LNCS 4390 (2007) 51–65
7. SO-Pharm. Online: [http://obofoundry.org/cgi-bin/detail.cgi?id=pharmacogenomics]
8. Smith, B., et al.: The obo foundry: coordinated evolution of ontologies to support biomedical data integration. Nat. Biotech. **25**(11) (2007) 1251 – 1255
9. Tomalik-Scharte, D., Lazar, A., Fuhr, U., Kirchheiner, J.: The clinical role of genetic polymorphisms in drug-metabolizing enzymes. Pharmacogenomics J. **8** (2008) 4–15
10. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook Cambridge University Press (2003)
11. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc., Secaucus, USA (1997)
12. Carpineto, C., Romano, G.: Concept Data Analysis: Theory and Applications. John Wiley & Sons (2004) 141 – 174
13. Ganter, B.: Two Basic Algorithms in Concept Analysis. Technische Hochschule, Darmstadt (1984)
14. Kryszkiewicz, M.: Concise representations of association rules. In: Proc. of the ESF Exploratory Workshop on Pattern Detection and Discovery. (2002) 92–109
15. Szathmary, L.: Symbolic Data Mining Methods with the Coron Platform. PhD Thesis in Computer Science, Nancy University, France (2006)
16. Kuznetsov, S.O., Obiedkov, S.A.: Comparing performance of algorithms for generating concept lattices. J. Exp. Theor. Artif. Intell. **14**(2-3) (2002) 189–216
17. Szathmary, L., Napoli, A., Valtchev, P.: Towards Rare Itemset Mining. In: ICTAI'07. (2007) 305–312
18. Hacene, M.R., Huchard, M., Napoli, A., Valtchev, P.: A proposal for combining formal concept analysis and description logics for mining relational data. In: Proc. of ICFCA. LNCS 4390 (2007) 51–65
19. Coulet, A.: Role Assertion Analysis: technical report. Technical report, INRIA Nancy - Grand Est, France (2008) Online: [http://www.loria.fr/~coulet/raa/technical_report.pdf]