Lexico-Logical Acquisition of OWL DL Axioms*

An Integrated Approach to Ontology Refinement

Johanna Völker and Sebastian Rudolph

Institut AIFB, Universität Karlsruhe, Germany
{voelker,rudolph}@aifb.uni-karlsruhe.de

Abstract. In order to overcome human and time resource problems in the task of ontology design, we propose to combine the LExO approach to learning expressive ontology axioms from textual definitions with Relational Exploration – a technique based on the well-known attribute exploration algorithm from FCA which is used to interactively clarify underspecified logical dependencies. By forcing particular modeling decisions the exploration of classes and class extension relationships guarantees completeness with respect to a certain logical fragment and increases the overall quality of the ontology. Providing an implementation as well as an example, we demonstrate how ontology learning and exploration complement each other in a synergetic way.

1 Introduction

In the prospering Semantic Web research field, ontologies – logical domain specifications useful for automatically drawing conclusions about the described domain – have taken a central role. Yet, building ontologies is a difficult and time-consuming task, requiring to combine the knowledge of domain experts with the skill and experience of ontology engineers resulting in a high demand on scarce expert resources. Moreover, the size of knowledge bases needed in real world applications easily exceeds the modeling capabilities of any human expert. On the other hand, both quality and expressivity of the ontologies generated automatically by the state-of-the-art ontology learning systems fail to meet the expectations of people who argue in favor of powerful, knowledge-intensive applications based on ontological reasoning.

In order to overcome this bottleneck, it is necessary to thoroughly assist the modeling process by providing hybrid semi-automatic methods which (i) intelligently suggest potentially relevant knowledge elements (complex domain axioms or facts) extracted from resources such as domain relevant text corpora and (ii) provide guidance during the knowledge specification process by asking decisive questions in order to clarify still undefined parts of the knowledge base.

Obviously, those two requirements complement each other. The first one clearly falls into the area of natural language processing. By using existing methods for knowledge extraction from texts, passages can be identified which indicate the validity of

* This work has been supported by the European Commission under contract IST-2006-027595 NeOn, and by the Deutsche Forschungsgemeinschaft (DFG) under the ReaSem project.

certain pieces of knowledge. For the second requirement, strictly logic-based exploration techniques such as the the well-known and well-established attribute exploration from formal concept analysis (and its variants and extensions) are needed in order to obtain logically crisp propositions. We believe that integrating these two directions of knowledge acquisition in one scenario will help overcoming disadvantages of either approach. The framework proposed in this paper realizes this integration and shows its potential for practical applications.

In Section 2, we briefly introduce the description logic $\mathcal{SHOIN}$. Section 3, sketches the field of ontology learning before presenting LExO as one method for acquiring DL axioms from texts. Section 4 gives the necessary background for Relational Exploration (RE), a technique used for interactive knowledge specification based on FCA. In Section 5, we describe in detail how LExO and RE (possibly assisted by other ontology learning components) can be synergetically combined in the process of ontology engineering and evaluation. Implementation details as well as an example are given in Section 6. Finally, Section 7 concludes and gives an outlook to future research.

2 Preliminaries

Here, we will very briefly introduce the description logic $\mathcal{SHOIN}$. A $\mathcal{SHOIN}$ knowledge base (KB, also: ontology) is based on sets N_R (*role names*) $\mathbf{C}$ (*atomic concepts*) and $\mathbf{I}$ (*individuals*). The set of $\mathcal{SHOIN}$ *roles* is $\mathbf{R} = \mathsf{N}_R \cup \{R^- \mid R \in \mathsf{N}_R\}$. In the following, we leave this vocabulary implicit and assume that A, B are atomic concepts, a, b, i are individuals, and R, S are roles. Those can be used to define concept descriptions employing the constructors from the upper part of Table 1. We use C, D to denote concept descriptions. Moreover, a $\mathcal{SHOIN}$ KB consists of two finite sets of axioms that are referred to as *TBox* and *ABox*. The possible axiom types for each are displayed in the lower part of Table 1.

Note that we do not explicitly consider concept or role equivalence $\equiv$, since it can be modeled via mutual concept or role inclusions. We adhere to the common model-theoretic semantics for $\mathcal{SHOIN}$ with general concept inclusion axioms (GCIs): an interpretation $\mathcal{I}$ consists of a set Δ called *domain* together with a function $\cdot^{\mathcal{I}}$ mapping individual names to elements of Δ, class names to subsets of Δ, and role names to subsets of $\Delta \times \Delta$. This function is inductively extended to roles and concept descriptions and finally used to decide whether the interpretation satisfies given axioms (see Table 1).

$\mathcal{SHOIN}$ serves as the theoretical basis for the web ontology language OWL DL as defined in [1]. OWL DL constitutes a standardized knowledge representation language well established in the Semantic Web domain. It is a fragment of first order predicate logic with the advantage of being decidable and even the availability of optimized reasoners for it.

3 Lexical and Logical Knowledge Acquisition

Ontology generation from natural language text, or lexical resources – most commonly referred to as "ontology learning" – is a relatively new field of research which aims to support the tedious task of knowledge acquisition by automatic means.

Name	Syntax	Semantics	
inverse role	R^-	$\{(x,y) \mid (y,x) \in R^{\mathcal{I}}\}$	
top	$\top$	Δ	
bottom	$\bot$	$\emptyset$	
nominal	$\{i\}$	$\{i^{\mathcal{I}}\}$	
negation	$\neg C$	$\Delta \setminus C^{\mathcal{I}}$	
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	
universal restriction	$\forall R.C$	$\{x \mid (x,y) \in R^{\mathcal{I}}$ implies $y \in C^{\mathcal{I}}\}$	
existential restriction	$\exists R.C$	$\{x \mid$ for some $y \in \Delta,\ (x,y) \in R^{\mathcal{I}}, y \in C^{\mathcal{I}}\}$	
(unqualified) number	$\leq n\, R$	$\{x \mid \#\{y \in \Delta \mid (x,y) \in R^{\mathcal{I}}\} \leq n\}$	
restriction	$\geq n\, R$	$\{x \mid \#\{y \in \Delta \mid (x,y) \in R^{\mathcal{I}}\} \geq n\}$	
role inclusion	$S \sqsubseteq R$	$S^{\mathcal{I}} \subseteq R^{\mathcal{I}}$	TBox
transitivity	$\mathsf{Trans}(S)$	$S^{\mathcal{I}}$ is transitive	TBox
general concept inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$	TBox
concept assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$	ABox
role assertion	$R(a,b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$	ABox

Table 1. Role/concept constructors and axiom types in $\mathcal{SHOIN}$. Semantics refers to an interpretation $\mathcal{I}$ with domain Δ. As usual, we require to restrict number restrictions to simple roles, i.e. (roughly speaking and omitting further technical details) roles that do not include roles which are declared to be transitive.

However, many of today's ontology learning approaches build upon methods and ideas which were developed by (computational) linguists long before ontologies became a popular means of knowledge representation. Ontology learning techniques based, e.g., on lexico-syntactic patterns [2], or Harris' distributional hypothesis [3] draw from previous advances in lexical acquisition, and terminology research which have been to a major extent focusing on the extraction of lexical relations. However, there is a tacit agreement in the ontology learning community that there exists a certain correspondence between lexical relations (e.g. hyponymy, synonymy), and ontological axioms (e.g. subsumption, equivalence). This assumption which is not only prevalent in ontology learning, but also influences manual ontology engineering[1] led to a kind of "lexical", i.e., lexically inspired ontology generation implemented in frameworks such as OntoLearn [4], OntoLT [5] or Text2Onto [6].

One may argue that due to the differences between lexical semantics, and the model-theoretic semantics of description logics (see also [7]), this type of approach will always yield at best light-weight, semi-formal ontologies without precisely defined semantics, being grounded in natural language more than in logics. On the other hand, lexical approaches to ontology generation offer a lot of advantages: They can benefit from large amounts of lexical resources such as machine-readable dictionaries, encyclopedias, and all kinds of web documents that are available in abundance on the web. The resulting ontologies are usually close to the human way of modeling, since they provide lexicalizations of classes, individuals and properties, thus being easily comprehensible and

[1] In fact, if one tries to explain the semantics of subsumption to a non-logician, one often resorts to "clue phrases" similar to lexico-syntactic patterns which themselves reflect lexical relations.

reusable. Finally, most of these approaches are very flexible with respect to the degree of user interaction, and relatively easy to combine with other, complementary or supporting ontology learning methods.

Besides those lexical methods, a second direction of ontology learning has received more and more attention during the last couple of years. Approaches based on Inductive Logic Programming (ILP) [8, 9] and Formal Concept Analysis (FCA) [10] have been developed in the logics community, for some reason widely unappreciated by lexical ontology learning research. Although there are a few approaches aiming to reconcile the two worlds by using either FCA [11, 12] or ILP [13] for lexical ontology acquisition, none of them has been designed specifically for the refinement of OWL DL ontologies or knowledge bases. Common to all those approaches is their idea to acquire knowledge based on presented domain entities and their properties. However, this type of logical ontology generation is often less efficient than lexical approaches, and requires a relatively large amount of manually acquired knowledge (e.g. ABox statements for taxonomy induction). The resulting ontologies lack the traceability of a natural language grounding, and meaningful labels for complex class descriptions. Their expressivity is typically restricted to some variant of $\mathcal{ALC}$. On the other hand, those approaches have several advantages. Since they are based on already structured, formal data, they naturally come with a precisely defined, formal set-theoretic semantics. Thus being on "safe logical grounds", it is guaranteed that the acquired knowledge is also logically consistent.

Despite their respective advantages, both lexical and logical approaches to automatic (or semi-automatic) ontology engineering have failed to meet all the expectations of people arguing in favor of knowledge-intensive, reasoning-based applications, e.g., in domains such as bio-informatics or medicine. In particular, expressivity and quality of the resulting axiomatizations are often insufficient for practical use. In order to meet these fundamental requirements, a few lexical approaches towards learning more expressive ontologies, i.e. ontologies featuring the expressiveness of OWL DL, have been proposed recently [7, 14]. But these approaches have to face a lot of challenges which need to be overcome in order to make them useable in practice. Obviously, the more expressive learned (or manually engineered) ontologies become, the more important it will be to provide automatic support for quality assurance, since the difficulty of a purely manual revision rises with the growing complexity of the ontology. On the other hand, applications relying on reasoning over complex ontologies make it necessary to consider a larger variety of qualitative aspects which must be taken into account as an ontology is being learned or constructed, including logical consistency, and completeness. Notwithstanding, there exist only very few frameworks aiming at a tight integration of methods for ontology learning and evaluation. Although, e.g., Haase et al. [15] propose a way to deal with logical inconsistencies in lexically generated ontologies the problem of modeling completeness has been largely neglected up to now.

In this paper, we therefore present an approach to ontology acquisition which effectively combines the strengths of the two complementary directions of research while at the same time compensating for many of their respective disadvantages. It relies upon Relational Exploration, an FCA-based approach to systematic, logical refinement (cf. Section 4), and the automatic generation of formal class descriptions by means of natu-

ral language processing techniques which is described in the remainder of this Section.

LExO[2] (Learning EXpressive Ontologies) [7] is an approach towards the automatic generation of ontologies featuring the expressiveness of OWL DL. The core of LExO is a syntactic transformation of definitory natural language sentences into description logic axioms. Given a natural language definition of a class, LExO starts by analyzing the syntactic structure of the input sentence. The resulting dependency tree is then transformed into a set of OWL axioms by means of manually engineered transformation rules. Possible input resources for LExO include all kinds of definitory sentences, i.e. universal statements about concepts, that can be found in online glossaries such as Wikipedia[3], comments in the ontology, or simply given by a domain expert.

In order to exemplify the approach, we assume that we would like to refine the description of the class *Reviewer* the semantics of which could be informally described as follows: *A reviewer is a person who reviews a paper that has been submitted to a conference or workshop.*[4] We will come back to this example in Section 6.

A minimum set of rules for translating this sentence into a DL class description is given by Table 2 (for a more complete listing of possible transformation rules and further explanations see [7]).

Rule	Natural Language Syntax	OWL Axioms
Disjunction	X: NP_0 `or` NP_1	$X \equiv (NP_0 \sqcup NP_1)$
Copula	X: NP_0 VBE NP_1	$NP_0 \equiv NP_1$
Relative Clause	X: NP_0 C(*rel*) VP_0	$X \equiv (NP_0 \sqcap VP_0)$
Verb with Prep. Compl.	X: V_0 $Prep_0$ NP(*pcomp-n*)$_0$	$X \equiv \exists V_0\_Prep_0.NP_0$

Table 2. Transformation Rules for *Reviewer*

Depending on the concrete set of translation rules and modeling preferences of the user, a translation of this sentence into OWL DL could then yield the following axioms:

reviewer
$\equiv$ *a_person_who_reviews_a_paper_that_has_been_submitted_to_a_conference_or_workshop*
a_person_who_reviews_a_paper_that_has_been_submitted_to_a_conference_or_workshop
$\equiv$ *a_person* $\sqcap$ *reviews_a_paper_that_has_been_submitted_to_a_conference_or_workshop*
reviews_a_paper_that_has_been_submitted_to_a_conference_or_workshop
$\equiv \exists$*reviews.a_paper_that_has_been_submitted_to_a_conference_or_workshop*
a_paper_that_has_been_submitted_to_a_conference_or_workshop
$\equiv$ *a_paper* $\sqcap$ *has_been_submitted_to_a_conference_or_workshop*
has_been_submitted_to_a_conference_or_workshop
$\equiv \exists$*has_been_submitted_to.a_conference_or_workshop*
a_conference_or_workshop $\equiv$ (*a_conference* $\sqcup$ *workshop*)

[2] `http://ontoware.org/projects/lexo/`
[3] `http://en.wikipedia.org`
[4] Depending on the intended meaning of *Reviewer* other, broader definitions (e.g. covering reviews of journal articles, or research projects) might be more adequate, but we wanted to keep the example as simple as possible.

Obviously, the above set of axioms can be normalized, and turned into a semantically equivalent, unfolded, representation:

$$\mathit{Reviewer} \equiv \mathit{Person} \sqcap \exists \mathit{review}.(\mathit{Paper} \sqcap \exists \mathit{submitted\_to}.(\mathit{Conference} \sqcup \mathit{Workshop}))$$

While such a compact class description might be easier to grasp at first glance (at least for ontology engineers being familiar with logics), the first axiomatization obviously conveys a lot of additional information to the human reader. The fact that each part of the overall class description (e.g. *Conference* ⊔ *Workshop*) is associated with an equivalent atomic class (e.g. *a_conference_or_workshop*) makes completely transparent how this axiomatization was constructed, and at the same time provides the user with an intuitive explanation of the semantics of each class description. Further advantages of the extended axiomatization are discussed in Section 5.

4 Relational Exploration

In order to sketch relational exploration (RE, introduced in [16] and thoroughly treated in [10]), we first need to briefly recall some basic notions from FCA (see [17] for further reference).

A *(formal) context* $\mathbb{K}$ is a triple (G, M, I) with an arbitrary set G (called *objects*), an arbitrary set M (called *attributes*), and a relation $I \subseteq G \times M$ (called *incidence relation*). We read gIm as: "object g has attribute m." Furthermore, let $g^I := \{m \mid gIm\}$. An *implication* on an arbitrary set M is written $A \to B$ with $A, B \subseteq M$. It *holds* in a formal context $\mathbb{K} = (G, M, I)$, if for all $g \in G$ we have that $A \subseteq g^I$ implies $B \subseteq g^I$. We then write $\mathbb{K} \models A \to B$. A set $\mathfrak{I}$ of implications *entails* $A \to B$ if $A \to B$ holds in all contexts wherein all implications from $\mathfrak{I}$ hold.

An implication set $\mathfrak{I}$ will be called *non-redundant*, if for any $(A \to B) \in \mathfrak{I}$ we have that $\mathfrak{I} \setminus \{A \to B\}$ does not entail $A \to B$. $\mathfrak{I}$ will be called *complete* w.r.t. a context $\mathbb{K}$, if every implication $A \to B$ holding in $\mathbb{K}$ is entailed by $\mathfrak{I}$. $\mathfrak{I}$ will be called an *implication base* of $\mathbb{K}$ if it is non-redundant and complete. Since implication entailment is known to be decidable in linear time [18], the implication base allows fast handling of an implicational theory. The classical attribute exploration algorithm [19, 20] provides a method for efficiently determining an implicational base of a formal context that is only implicitly known by an expert.

The technique of RE extends this algorithm to a DL setting: Given an interpretation $\mathcal{I}$ on a domain Δ and a set M of $\mathcal{SHOIN}$ concept descriptions, the corresponding *$\mathcal{I}$-context* is defined by $\mathbb{K}_{\mathcal{I}}(M) := (\Delta, M, I)$ with $\delta I C :\Leftrightarrow \delta \in C^{\mathcal{I}}$. Then it can be easily shown, that implications in $\mathbb{K}_{\mathcal{I}}$ coincide with certain axioms w.r.t. their validity in $\mathcal{I}$: for $\mathcal{C}, \mathcal{D} \subseteq M$, the implication $\mathcal{C} \to \mathcal{D}$ holds in $\mathbb{K}_{\mathcal{I}}$ if and only if $\mathcal{I}$ satisfies the DL axiom $\bigsqcap \mathcal{C} \sqsubseteq \bigsqcap \mathcal{D}$. Hence it is possible to explore DL axioms (more precisely: general concept inclusion axioms, short: GCIs) with this technique. In an interview-like process, a domain expert has to judge whether a proposed GCI is valid in the domain (formally: the interpretation $\mathcal{I}$) he is describing and in the negative case provide a counterexample.[5] Since OWL DL [1] – the standard language for representing ontologies – is based on

[5] This will be further elaborated and demonstrated in the subsequent sections.

description logics, the RE method easily carries over to any kind of ontologies specified in that language.

Especially when working in an OWL or DL setting, the open world assumption is omnipresent; most of the known objects will not be completely specified, i.e., for certain classes it might be unknown whether the considered individual is an instance. Hence, it is essential for exploration methods to be capable of dealing with this kind of information. Lately, there has been significant work on applying FCA results on partial information (e.g. described in [21, 20]) to the ontology refinement setting. An according approach (briefly sketched in [10]) has been fully theoretically elaborated and implemented as described in [22]. It allows to use partly specified objects as counterexamples for hypothetical implications. We decided to follow this approach, hence the implementation presented in the remainder of this paper allows for handling partial contexts.

The advantage of RE is that the obtained results are logically crisp and naturally consistent. Moreover, the acquired information is complete with respect to certain well-defined logic fragments of OWL DL.[6] Yet, one major shortcoming of RE is the following: due to the aimed-at completeness, the number of asked questions (and therefore, the runtime and the workload for the expert) grows rapidly with the number of involved concepts and roles which threatens to exceed the ontology designers resources.

In order to counter this we propose a combination of two strategies: firstly, we use an OWL DL reasoner to determine whether the answer to a question posed by the exploration algorithm can be deduced from a previously given background knowledge ontology. Secondly we use lexical ontology learning to determine a relatively small number of relevant classes to focus on. Both points will be elaborated in the next section.

5 An Integrated Approach to Ontology Refinement

In the sequel, we will describe how LExO and RE can be synergetically combined by giving a comprehensive description of the integrated algorithm. En route, we will briefly mention how other lexical ontology learning techniques could be beneficially used within that process. In addition to the LExO and RE component, an OWL DL reasoner will be applied in order to draw conclusions that are already implicitly present, i.e. entailed by the actual knowledge base making an intervention of the user obsolete.

Creation of new Definitions and Mappings. We start with an OWL DL ontology $\mathcal{KB}$ to be refined with respect to a (new or already contained) class C, for which a natural language definition is provided by some textual resource. This textual definition is then analyzed by LExO yielding a set $\mathcal{KB}$' of OWL DL axioms as described in Section 3. Most likely, some (or even most) of the named classes those axioms refer to will not be present in $\mathcal{KB}$. Therefore, at least the primitive classes amongst those – i.e. those classes not stated to be equivalent to a complex class description[7] – should be linked to $\mathcal{KB}$. There are several ways for doing that. If textual definitions are available, LExO could be employed "recursively", i.e., it might be applied to the definitions of the classes in question in order to obtain other classes that can be linked to $\mathcal{KB}$ more easily.

[6] Which fragment precisely depends on which variant of RE is used.

[7] These are the classes occurring explicitly in the normal form (cf. Section 3).

In any case, ontology *mappings* between $\mathcal{KB}$ and $\mathcal{KB}'$ could be either added manually or established by one or several of the well-known mapping tools like FOAM[8] [23]. So let Map be a (possibly empty) set of respective mapping axioms.

Selection of Relevant Classes. In the next step, we stipulate the focus of the subsequent exploration, by selecting the named classes from $\mathcal{KB} \cup \mathcal{KB}'$ whose logical dependencies shall be further clarified. A natural default choice for this would be the set of all named classes from $\mathcal{KB}'$, as we might suppose the (remaining) classes from $\mathcal{KB}$ to be modeled in a sufficiently precise way – an assumption that might be disproved later on. However, it might be reasonable to include some of the classes from $\mathcal{KB}$ as well. Knowledge extraction methods that determine the relevance of terms (like those offered by Text2Onto [6]) could be employed for an automatic selection or to generate reasonable suggestions. In any case, let $\mathbf{C}$ denote the set of selected attributes.

After this selection of relevant named classes, a basic fact from FCA allows to further restrict $\mathbf{C}$: put into DL notation, it assures the dispensability of a class $C \in \mathbf{C}$ whenever there is a set $\mathbf{D} = \{D_1, \ldots, D_n\} \subseteq \mathbf{C} \setminus \{C\}$ such that $C \equiv D_1 \sqcap \ldots \sqcap D_n$ follows from all knowledge $\mathcal{KB}_\Sigma := \mathcal{KB} \cup \mathcal{KB}' \cup Map$ stated so far.[9] It takes just a little consideration that this is the case iff

$$\mathcal{KB}_\Sigma \models \bigsqcap \left\{ D \;\middle|\; D \in \mathbf{C} \setminus \{C\},\ \mathcal{KB}_\Sigma \models C \sqsubseteq D \right\} \sqsubseteq C,$$

such that the elimination of redundant classes from $\mathbf{C}$ requires just $O(|\mathbf{C}|^2)$ reasoner calls in the worst case. Let $\mathbf{C}'$ denote the result of this reduction process.

Exploration. Now we start RE as described in Section 4 on the concept set $\mathbf{C}'$. A work flow diagram of the procedure is displayed in Figure 1. For every hypothetical DL axiom $C_1 \sqcap \ldots \sqcap C_n \sqsubseteq D_1 \sqcap \ldots \sqcap D_m$ brought up by the exploration algorithm:

- Employ the reasoner to check whether this GCI is a consequence of $\mathcal{KB}_\Sigma$. If so, confirm the implication and continue the exploration with the next hypothesis.
- Employ the reasoner to query for all individuals γ with $C_1 \sqcap \ldots \sqcap C_n \sqcap \neg D_i(\gamma)$ for an i from $1, \ldots, m$, i.e., for instances of the class which characterizes the property for being a material counterexample[10] for the hypothetical GCI. Let Γ be the set of individuals retrieved this way. If $\Gamma \neq \emptyset$, select one $\gamma \in \Gamma$ and check for every $C \in \mathbf{C}$ whether $C(\gamma)$ or $\neg C(\gamma)$. Then the counterexample together with the information about the attributes it provably has or has not is passed to the exploration algorithm. Optionally the human expert – possibly assisted by lexical knowledge retrieval tools – might be asked to complete the assertions for γ in order to get a more specific description for it. In any case, after providing γ, the exploration will proceed with the next hypothesis.

[8] `http://www.aifb.uni-karlsruhe.de/WBS/meh/foam/`

[9] In FCA terms this can be conceived as a kind of a-priori attribute reduction. Note that this process is nondeterministic. In case two classes happen to be equiextensional, we nondeterministically remove one of them.

[10] Material counterexamples are objects for which is known which part of the conclusion they violate. The exploration algorithm (even the one dealing with partial knowledge) can only make use of this kind of counterexamples.

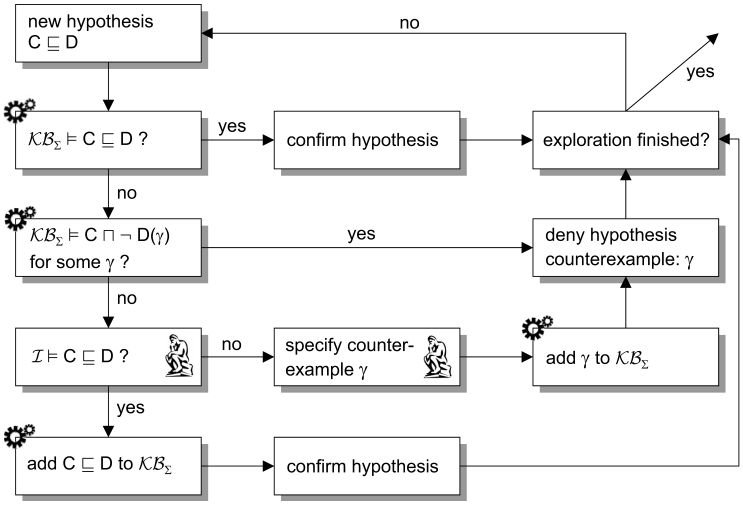

Fig. 1. Relational Exploration process (the gear wheels indicate ontology management activities including reasoning and updates, whereas the thinker icon marks user involvement).

– If the DL axiom in question can be neither automatically proved nor declined (the latter meaning $\Gamma = \emptyset$), the human will be asked for the ultimate decision whether the axiom is satisfied in the described domain $\mathcal{I}$ or not. Again, ontology learning tools could support him by suggesting answers endowed with a probability, or simply scanning a corpus for potential hints and presenting selected passages.

The exploration terminates after finitely many steps, yet it may also be stopped by the user beforehand. In the latter case, the internal order of the classes from the set $\mathbf{C}'$ is relevant since it determines the order of the posed questions. Hence, it is beneficial to sort those classes w.r.t. their relevance, possibly based on textual information. After the exploration cycle being finished, we have obtained a refined knowledge base $\mathcal{KB}_\Sigma$ containing the (possibly new) class C endowed with its definition (as extracted from the textual definition) and its interrelationships with concepts from the original knowledge base. Additionally, the "semantic neighborhood" of C has been made logically explicit by interactive exploration. In fact, any subsumption between conjunctions of classes from $\mathbf{C}$ can be decided (i.e. proven or disproven) based on the refined knowledge base. This also shows the advantage of introducing atomic classes for the complex concept descriptions occurring in the LExO output as demonstrated in Section 3: although RE as applied in this case[11] deals only with conjunctions on atomic classes, we introduce more expressivity "through the back-door" by having complex definitions for those named classes in our ontological background ready to be exploited by the reasoner.

The synergies provided by the presented combination are manifold: Firstly, the classes contained in the definitions provided by LExO provide a reasonable small to medium size "exploration scope" being crucial for a reasonable application of the RE technique. Secondly, we can use textual information for generating ontological information (a source not accessible to purely logical approaches) yet being able to interactively

[11] Actually, RE provides means for exploring GCIs in whole $\mathcal{ALE}$ with bounded role depth, however we restrict to conjunctions on atomic classes in this example.

clarify logical dependencies that have been left open by the text. The latter is done in a guided way ensuring completeness.

Overall, the proposed framework provides means for interactively integrating learned or manually acquired axiomatizations into an existing ontology, while at the same time facilitating their evaluation and refinement.

6 Implementation and Example

In order to prove the feasibility of a synthesis of ontology learning and RE as described in Section 5, we implemented a prototypical application named RELExO. Both sources and binaries of RELExO are available for public use and can be downloaded from its homepage[12] which has been set up to provide further information with respect to our experiments on ontology learning and relational exploration. RELExO relies upon KAON2[13] as an ontology management back-end and features a simple graphical user interface. Its architecture is depicted by Figure 2.

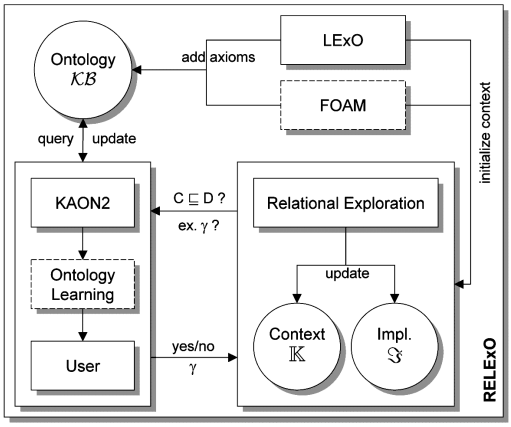

Fig. 2. RELExO Architecture

LExO, possibly complemented by other ontology learning components, generates or extends the initial set of axioms $\mathcal{KB}$ (mappings can be added by FOAM, if necessary), and initializes the partial context $\mathbb{K}$ by suggesting a set of attributes $\mathbf{C}$ to the user. The actual refinement process is handled by a RE component which manages the partial context $\mathbb{K}$ and the implication set $\mathfrak{I}$. Both are updated based on answers obtained from the "expert team" constituted by the KAON2 reasoner, an optional ontology learning component as well as the human knowledge engineer.

We now illustrate the integrated ontology refinement process which has been elaborated on in Section 5 by means of a real-world example. The complete material necessary for reproducing this example, i.e. ontologies and screenshots, is contained in the RELExO distribution.

[12] `http://relexo.ontoware.org`
[13] `http://kaon2.semanticweb.org`

The **SWRC** (Semantic Web for Research Communities)[14] [24] ontology is a well-known ontology modeling the domain of Semantic Web research. Version 0.7 contains 71 classes, e.g., for different types of persons, publication, and events, 48 object properties, 46 datatype properties, and an overall number of 672 axioms. Its expressiveness is slightly beyond OWL DLP featuring subsumption, properties, and a few disjointness axioms. The ontology serves as a basis for semantic annotation in the AIFB web portal[15] which manages information about more than 2,000 persons, projects, and publications. For the purpose of our experiment, we exported all instance data stored in the AIFB portal into one single OWL file (more than 3 Megabytes in RDF syntax), and merged it with the corresponding TBox, i.e. the latest version of SWRC. After minor syntactic corrections (removing non XML-compliant characters), we obtained a considerably large ontology. Debugging with RaDON[16] revealed two inconsistencies caused by conflicting range specifications of data properties which could be fixed without difficulty.

Subsequently (in order to keep the example simple and rule out a few trivial questions that would otherwise come up in the exploration phase), we added axioms stating the disjointness of the SWRC top-level concepts *Person*, *Event*, and *Publication* – obviously true axioms yet not present in the current version of this ontology. Those axioms could also have been generated automatically by techniques for learning disjointness from [14]. However, adding these axioms turned the ontology inconsistent again as some individuals were inferred to instantiate both *Person* and *Publication*. The reason for this inconsistency was an incorrect use of the *editor* relationship in SWRC. Although its domain was restricted to *Person* ("*editor_of*"), the property was apparently conceived to have "*has_editor*" semantics by most of the annotators. We fixed this inconsistency by changing the definition of *editor* accordingly. Another problem became apparent after we had already started the exploration of the resulting ontology with RELExO. An individual (in our opinion) belonging to the class *ResearchPaper* was proposed as a counterexample, but could not be classified as a such. A closer look at both individual and ontology showed that it was assigned to the class *InProceedings* which was declared disjoint from *ResearchPaper*, the latter actually being empty. Since we found that this modeling decision is not justified by the associated comments in the ontology, we simply removed the disjointness axiom.

To demonstrate the RELExO approach, we assume that we would like to add a new class *Reviewer* to the SWRC ontology. Part of a change request could be a natural language description of this class such as *"a reviewer is a person who reviews a paper that has been submitted to a conference or workshop"* (cf. Section 3). Given this definitory sentence, LExO automatically suggests an axiomatization of *Reviewer* to the user who can correct or remove some of the generated axioms before they are added to the ontology. Applying FOAM for suggesting mappings between the newly introduced class names and those already present in SWRC, we find *Paper* to be equivalent to *ResearchPaper* and add a corresponding equivalence axiom to the extended ontology. Likewise, we find *Person*, *Conference* and *Workshop* already present in the original ontology.

[14] `http://ontoware.org/projects/swrc/`
[15] `http://www.aifb.uni-karlsruhe.de`
[16] `http://radon.ontoware.org`

In the next step, the set of "relevant" classes has to be selected. As mentioned in Section 5, it is reasonable to choose those atomic classes present in the definition of *Reviewer*. We decided to add two more classes denoting undergraduate and PhD students and (introducing abbreviations for overly long concept names from $\mathcal{KB}$') we set:

$$\mathbf{C}' := \{\bot, \mathit{CoW}, \mathit{Conference}, \mathit{SubCoW}, \mathit{Person}, \mathit{PhDStudent}, \\ \mathit{ResearchPaper}, \mathit{RevPSubCoW}, \mathit{Undergraduate}, \mathit{Workshop}\}.$$

Based on this set of classes, the RE algorithm is started. The first hypothetical DL axiom, the exploration comes up with is $\top \sqsubseteq \bot$. Naturally, this hypothesis cannot be deduced from the ontology. Hence, following the description in Section 5, KAON2 will query the knowledge base for instances of $\top \sqcap \neg\bot$ which is equivalent to $\top$. Hence *all* ABox individuals are retrieved. Choosing one of the retrieved individuals, in our case *id1289instance*, we find it to be an instance of *ResearchPaper* and (since in our example,we chose the option to give the expert the opportunity to enhance the counterexample specification) add the information that it is an instance of *SubCoW*.

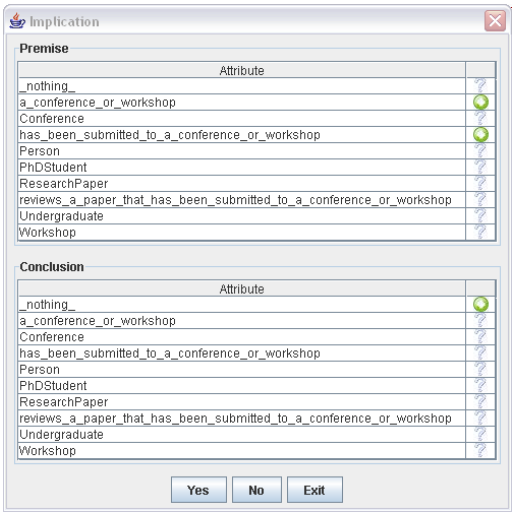

Fig. 3. Dialog for displaying the hypothetical axiom $\mathit{CoW} \sqcap \mathit{SubCoW} \sqsubseteq \bot$.

In a similar way, the next hypothesis posed – $\top \sqsubseteq \mathit{ResearchPaper} \sqcap \mathit{SubCoW}$ – is handled. Clearly, not every ABox individual is a research paper witnessed by the counterexample *id1303instance* being a journal article and hence neither a research paper (according to the underlying ontology) nor submitted to a conference or workshop.

However, the subsequent hypothesis $\mathit{CoW} \sqsubseteq \bot$ can neither be proved nor disproved by KAON2 using the information actually present in the ontology – since it does not contain any individuals being a conference or workshop. Therefore, the human expert will be asked for the final decision. Obviously, this hypothesis has to be denied and a counterexample for it is just any conference, so we enter *ICFCA_2008* and specify

it as instance of *Conference*.[17] Note that due to the capability of dealing with partial information, the expert may leave open whether this individual belongs to the other considered classes. However, we employ the reasoner in order to determine all class-memberships deducible from the present information. In our case, it can be inferred that *ICFCA_2008* is also an instance of *CoW* and definitely no instance of *ResearchPaper*.

Consequently, the next question $CoW \sqsubseteq Conference$ comes up and has to be denied as well by entering the workshop instance *OntoLex_2007*.

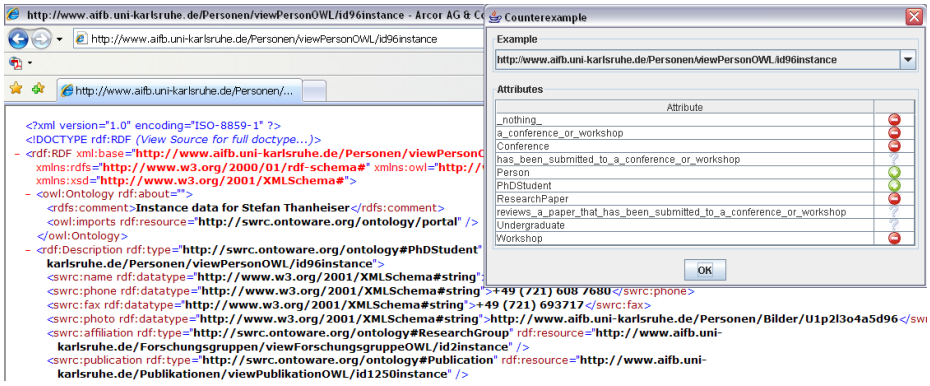

Fig. 4. Specifying a counterexample. Every (non-)class-membership deducible from the knowledge base is automatically entered leaving just the open questions to the expert. RELExO can be configured to automatically display the web page associated with an individual's URI.

Equally, the hypothesis $SubCoW \sqsubseteq ResearchPaper$ cannot be decided based on the present knowledge and is thus passed to the expert. In fact, this is the first "design decision" to make depending on the intended scope of the ontology. A look into the SWRC taxonomy reveals that there is a class *Poster* to denote posters presented at conferences. Indeed, any submitted poster would be a counterexample for the presented hypothesis, so we add *iMapping_Poster_SWUI_2006* to the knowledge base.

The next hypothesis brought up is $CoW \sqcap SubCoW \sqsubseteq \bot$ being an integrity constraint saying that nothing being a conference or workshop can be submitted (to a conference or workshop). Figure 3 shows how it is presented to the user. Here, we encounter another design decision. Although it might be reasonable to say that a workshop (actually: a workshop proposal) has been submitted to a conference, we stick to the intended semantics of the term *Workshop* as a kind of event which cannot be submitted and hence confirm the validity of the presented hypothesis.

The hypothesis $Person \sqsubseteq \bot$, coming up next, is refuted by the reasoner retrieving an individual who is a PhD student at the institute AIFB. Figure 4 shows the dialog wherein the user is presented the stored information about this individual and is asked to add the missing facts.

[17] This information already qualifies *ICFCA_2008* as a counterexample for the presented hypothesis. RELExO checks for every alleged counterexample whether it is indeed a such and rejects the input otherwise.

In this way, the exploration continues. During the process, some individuals are added and the following new axioms are confirmed:

- $SubCoW \sqcap Person \sqsubseteq \bot$ (a person cannot be submitted)
- $ResearchPaper \sqsubseteq SubCoW$ (every research paper has been submitted to a conference or workshop)[18]
- $RevPSubCoW \sqsubseteq Person$ (everybody reviewing a submitted paper is a person)
- $Person \sqcap PhDStudent \sqcap Undergraduate \sqsubseteq \bot$ (PhD students are disjoint with undergraduates)[19]
- $RevPSubCoW \sqcap Undergraduate \sqcap Person \sqsubseteq \bot$ (actually a "policy decision": undergraduates are not allowed to review papers)

Fig. 5. Partial formal context resulting from the exploration.

The formal context with the examples acquired during the exploration is displayed by Figure 5. It is automatically exported to the native ConExp[20] format and stored as a `CEX` file.

We end up with a refined SWRC ontology containing the new class *Reviewer* fully integrated into the existing ontology. Any subsumption between conjunctions of the specified interesting classes can be directly decided based on this refined SWRC ontology. This can be nicely demonstrated by starting RELExO again with the refined ontology: it terminates without ever asking the human expert for a decision, showing that all upcoming questions can be answered by the reasoner alone.

7 Conclusion and Outlook

In this paper, we have sketched a way to combine complementary approaches to ontological knowledge acquisition: the more intensional approach of distilling conceptual information from textual resources, and the extensional method of extracting hypothetical domain axioms based on given entities. We have instantiated this approach by designing and implementing a framework that integrates the LExO ontology learning application, a Relational Exploration component, and the KAON2 reasoner. To the best of our knowledge, RELExO is the first publicly available implementation of an exploration-based ontology refinement approach. It is open source and supports the standard ontology language OWL. In an

[18] We regard this justified by the existence of a class *Unpublished* disjoint to *ResearchPaper*.
[19] Another modeling flaw: this axiom should have been present in SWRC.
[20] http://conexp.sourceforge.net

example using the well-known SWRC ontology we have demonstrated the feasibility of our approach, and its applicability to real-world ontology engineering tasks.

Altogether, we are confident that the proposed framework will considerably alleviate the task of designing comprehensive and complex, yet logically consistent ontologies for knowledge-intensive applications. The number of design decisions to be made by the human user is minimized by the usage of textual resources and the employment of a reasoning back-end. Relational exploration provides guidance, ensuring that neither redundant information will be asked for nor important information is simply forgotten in the modeling process, and supports on-the-fly ontology evaluation: as in our example (where we were well-nigh inevitably confronted with design flaws in the used ontology), present modeling errors in the ontology are often indicated by "surprising" or counterintuitive questions asked by the algorithm. Hence from the methodological point of view, a cyclic ontology engineering process with intertwined exploration and manual refinement (or debugging) phases seems a promising strategy.

After all, human intervention will always remain indispensable, especially for complex knowledge modeling tasks. Notwithstanding, the workload to ontology engineers and domain experts can be drastically decreased by intelligently integrated components for semi-automatic ontology engineering. By facilitating the acquisition of expressive OWL ontologies, we hope to foster the development of more sophisticated, reasoning-based applications, and help to put semantic technologies into practice.

Pursuing this promising goal, we identify several central issues for future research. Firstly, we are planning to incorporate the just recently proposed technique of role exploration from [25]. In order to achieve an even tighter lexico-logical integration, the implementation of RELExO could be further extended by an additional (automatic) expert which uses ontology learning techniques, and online resources for confirming hypotheses, or suggesting counterexamples. Additional ontology learning components could be used to complement the LExO-generated axiomatizations by other modeling primitives (e.g. disjointness axioms), or to sort the attributes, i.e. class descriptions, with respect to the current domain or the user's interests. Finally, we will integrate RELExO into an ontology engineering environment such as the NeOn Toolkit[21], and improve its usability by adding a natural language generation component for translating hypotheses, i.e. logical implications, into natural language questions. In the end, we are confident that further extensive evaluations in real world application scenarios will demonstrate the advantages of our combined, lexico-logical approach.

References

1. McGuinness, D.L., van Harmelen, F.: OWL Web Ontology Language Overview. `CTAN: http://www.w3.org/TR/2004/REC-owl-features-20040210/` (2004)
2. Hearst, M.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th International Conference on Computational Linguistics. (1992) 539–545
3. Harris, Z.S.: Word. Distributional Structure **10** (1954) 146–162
4. Velardi, P., Navigli, R., Cuchiarelli, A., Neri, F.: Evaluation of ontolearn, a methodology for automatic population of domain ontologies. In: Ontology Learning from Text: Methods, Applications and Evaluation. IOS Press (2005)

[21] `http://www.neon-toolkit.org`

5. Buitelaar, P., Olejnik, D., Sintek, M.: OntoLT: A Protégé plug-in for ontology extraction from text. In: Proceedings of the International Semantic Web Conference (ISWC). (2003)
6. Cimiano, P., Völker, J.: Text2Onto - a framework for ontology learning and data-driven change discovery. In: Proc. of the 10th International Conference on Applications of Natural Language to Information Systems, Springer (2005) 227–238
7. Völker, J., Hitzler, P., Cimiano, P.: Acquisition of OWL DL axioms from lexical resources. In: Proc. of the 4th European Semantic Web Conference (ESWC'07), Springer (2007)
8. Fanizzi, N., Iannone, L., Palmisano, I., Semeraro, G.: Concept formation in expressive description logics. In: Proc. of the 15th European Conference on Machine Learning (ECML'04), Springer Verlag (2004)
9. Cohen, W.W., Hirsh, H.: Learning the classic description logic: Theoretical and experimental results. In Doyle, J., Sandewall, E., Torasso, P., eds.: Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR'94). Bonn, Germany, May 24-27, 1994, Morgan Kaufmann (1994) 121–133
10. Rudolph, S.: Relational Exploration - Combining Description Logics and Formal Concept Analysis for Knowledge Specification. Universitätsverlag Karlsruhe (2006) Dissertation.
11. Stumme, G., Maedche, A.: FCA-merge: Bottom-up merging of ontologies. In: Proc. 17th International Conference on Artificial Intelligence (IJCAI '01). (2001) 225–230
12. Cimiano, P., Hotho, A., Staab, S.: Learning concept hierarchies from text corpora using formal concept analysis. Journal of Artificial Intelligence Research **24** (2005) 305–339
13. Nedellec, C.: Corpus-based learning of semantic relations by the ILP system, asium. In Cussens, J., Dzeroski, S., eds.: Learning Language in Logic. Volume 1925 of Lecture Notes in Computer Science., Springer (1999) 259–278
14. Völker, J., Vrandecic, D., Sure, Y., Hotho, A.: Learning disjointness. In: Proc. of the 4th European Semantic Web Conference (ESWC'07), Springer (2007)
15. Haase, P., Völker, J.: Ontology learning and reasoning - dealing with uncertainty and inconsistency. In da Costa, P.C.G., Laskey, K.B., Laskey, K.J., Pool, M., eds.: Proc. of the Workshop on Uncertainty Reasoning for the Semantic Web (URSW). (2005) 45–55
16. Rudolph, S.: Exploring relational structures via FLE. In Wolff, K.E., Pfeiffer, H.D., Delugach, H.S., eds.: Conceptual Structures at Work: 12th International Conf. on Conceptual Structures. Volume 3127 of LNCS., Huntsville, AL, USA, Springer (2004) 196 – 212
17. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1997) Translator-C. Franzke.
18. Maier, D.: The Theory of Relational Databases. Computer Science Press (1983)
19. Ganter, B.: Two basic algorithms in concept analysis. Technical Report 831, FB4, TH Darmstadt (1984)
20. Ganter, B.: Attribute exploration with background knowledge. Theoretical Computer Science **217** (1999) 215–233
21. Burmeister, P.: Merkmalimplikationen bei unvollständigem Wissen. In Lex, W., ed.: Arbeitstagung Begriffsanalyse und Künstliche Intelligenz, TU Clausthal (1991) 15–46
22. Baader, F., Ganter, B., Sertkaya, B., Sattler, U.: Completing description logic knowledge bases using formal concept analysis. In Veloso, M.M., ed.: IJCAI. (2007) 230–235
23. Ehrig, M., Sure, Y.: FOAM - framework for ontology alignment and mapping. results of the ontology alignment initiative. In Ashpole, B., Ehrig, M., Euzenat, J., Stuckenschmidt, H., eds.: Proc. of the Workshop on Integrating Ontologies. Volume 156. (2005) 72–76
24. Sure, Y., Bloehdorn, S., Haase, P., Hartmann, J., Oberle, D.: The SWRC ontology - semantic web for research communities. In Bento, C., Cardoso, A., Dias, G., eds.: Proc. of the 12th Portuguese Conference on Artificial Intelligence, Springer (2005) 218 – 231
25. Rudolph, S.: Acqiring generalized domain-range restrictions. In: Proc. of the 6th International Conference on Formal Concept Analysis, Springer (2008) to appear.