A Modularity Approach for a Fragment of $\mathcal{ALC}$

Andreas Herzig and Ivan Varzinczak

IRIT – 118 route de Narbonne
31062 Toulouse Cedex – France
{herzig,ivan}@irit.fr
http://www.irit.fr/LILaC

Abstract. In this paper we address the principle of modularity of ontologies in description logics. It turns out that with existing accounts of modularity of ontologies we do not completely avoid unforeseen interactions between module components, and modules designed in those ways may be as complex as whole theories. We here give a more fine-grained paradigm for modularizing descriptions. We propose algorithms that check whether a given terminology is modular and that also help the designer making it modular, if needed. Completeness, correctness and termination results are demonstrated for a fragment of $\mathcal{ALC}$. We also present the properties that ontologies that are modular in our sense satisfy w.r.t. reasoning services.

Keywords: Knowledge representation, description logics, modularity.

1 Motivation

Imagine an automatic passport control system in an airport such that all passengers must be controlled. Besides other software components, such a system is built on a passenger ontology. Suppose that the ontology is made up of statements like "a passenger has a passport", "EU citizens have EU passports", and "foreigners have non-EU passports". Such a knowledge can be encoded in description logics like $\mathcal{ALC}$ [1] by the following terminological axioms: $\mathsf{Passenger} \sqsubseteq \exists \mathsf{passport}.\top$, $\mathsf{EUcitizen} \equiv \forall \mathsf{passport}.\mathsf{EU}$, and $\mathsf{Foreigner} \equiv \forall \mathsf{passport}.\neg\mathsf{EU}$. Moreover, let the axiom $\mathsf{DoubleCitizen} \equiv \mathsf{Foreigner} \sqcap \mathsf{EUcitizen}$ define a foreigner that also has got a second citizenship of some EU country. It is not that hard to see that this description is consistent. Now, from such an ontology it follows $\mathsf{DoubleCitizen} \equiv \forall \mathsf{passport}.\bot$, and from this and the axiom $\mathsf{Passenger} \sqsubseteq \exists \mathsf{passport}.\top$ we conclude $\mathsf{DoubleCitizen} \sqsubseteq \neg\mathsf{Passenger}$, i.e., a person with double citizenship is not a passenger. Hence, if we have the assertion $\mathsf{DoubleCitizen}(\mathsf{BINLADEN})$, regarding the system behavior, this means that the concerned individual does not necessarily need to be controlled!

Despite the simplicity of such a scenario, problems like this are very likely to happen, especially if the knowledge base gets huge and hence more difficult to control. An alternative to ease maintainability of large ontologies is decomposing it into modules. Starting with [6], where modularity is assessed in logical theories in general, this issue has been investigated for ontologies in the recent

M. Fisher et al. (Eds.): JELIA 2006, LNAI 4160, pp. 216–228, 2006.

literature on the subject [15, 5]. Nevertheless, it turns out that these methods for modularizing descriptions, i.e., creating independent partitions of a knowledge base, do not take into account internal interactions of components of the description that can lead to unintuitive conclusions like the one above, even if the ontology is consistent. Here we go further and propose a more fine-grained modularity principle with which we get a decomposition of the ontology so that interactions between and inside their components are limited and controlled.

Ontologies are usually represented by DL knowledge bases containing multiple roles $R_1, R_2, \ldots$ Such roles are used to formalize attributes of a concept. Then we naturally have modularity whenever a given ontology description Σ can be partitioned into sub-descriptions relative to each role:

$$\Sigma = \Sigma^{\emptyset} \cup \Sigma^{R_1} \cup \Sigma^{R_2} \cup \ldots$$

such that

- $\Sigma^{\emptyset}$ contains no role references, and
- the only role of Σ^{R_i} is R_i.

We call these sub-descriptions *modules* (some modules might be empty). Examples of such modules can easily be found in design of DL ontologies, where each Σ^{R_i} contains axioms involving only the role R_i, and $\Sigma^{\emptyset}$ is the sub-description whose axioms mention no role at all, i.e., contains only boolean combinations of concepts.

For example, for our passport control system we have the description:

$$\Sigma^{\mathsf{passport}} = \left\{ \begin{array}{c} \mathsf{Passenger} \sqsubseteq \exists \mathsf{passport}.\top, \\ \mathsf{EUcitizen} \equiv \forall \mathsf{passport}.\mathsf{EU}, \\ \mathsf{Foreigner} \equiv \forall \mathsf{passport}.\neg \mathsf{EU} \end{array} \right\}$$

$$\Sigma^{\emptyset} = \{\mathsf{DoubleCitizen} \equiv \mathsf{Foreigner} \sqcap \mathsf{EUcitizen}\}$$

Such a description is composed of two sub-descriptions, one for expressing the *attributive* part of the theory, $\Sigma^{\mathsf{passport}}$, and one to formalize the role-free constraints of the domain, $\Sigma^{\emptyset}$. $\Sigma^{\mathsf{passport}}$ formalizes the restrictions on the attributes of the concepts of the domain, in this case that a passenger must have a passport, that an EU citizen has an EU passport, and so on. $\Sigma^{\emptyset}$ establishes the *boolean* constraint according to which a double citizen is a foreigner and an EU citizen, with no regard to his attributes.

A similar partitioning of descriptions can be found in reasoning about actions, where each Σ^{a} contains descriptions of the atomic action a in terms of preconditions and effects, and $\Sigma^{\emptyset}$ is the set of static laws (alias domain constraints), i.e., those formulas that hold in every possible state of a dynamic system, and are thus global axioms. Another example is when mental attitudes such as knowledge, beliefs or goals of several independent agents are represented: then each module Σ^{α} contains the respective mental attitudes of agent α.

Let Σ denote a description logic ontology and suppose we want to know whether $\Sigma \models C \sqsubseteq D$, i.e., whether an axiom $C \sqsubseteq D$ follows from the description in Σ. Then it is natural to expect that we only have to consider those modules of Σ which concern the alphabet of $C \sqsubseteq D$, more specifically the roles occurring in $C \sqsubseteq D$. For instance, deductions concerning the role $\mathsf{passport}$ should not involve axioms for role $\mathsf{hasDisease}$; querying the ontology of the passport control system should not require bothering with that of the fast-food in the airport hall. This is the problem we address in this paper.

The present work is structured as follows: in Section 2 we recall some logical definitions that we will use throughout this paper. In Section 3 we present a role-based decomposition of ontologies, which will serve as guideline for the definition of modularity in description logics we give in Section 4. We then define a fragment of $\mathcal{ALC}$ for which we have a sound and complete modularity test (Section 5). Before concluding, we show some of the benefits we get from ontologies that are modular in our sense (Section 6).

2 Description Logic $\mathcal{ALC}$

Here we briefly present the basic definitions of the description logic $\mathcal{ALC}$. For more details, see [1].

The basic syntactic building blocks of $\mathcal{ALC}$ as of any other description logics are atomic *concepts*, atomic *roles*, and *individuals*. We call atomic concepts and atomic roles elementary descriptions. Complex descriptions are built from them with concept constructors. We use A to denote atomic concepts, R for atomic roles, and $C, D, \ldots$ for complex concept descriptions.

Complex concept descriptions are recursively defined in the following way:

$$
\begin{array}{rll}
C ::= & A \mid & \text{(an atomic concept)} \\
& \top \mid & \text{(universal concept)} \\
& \bot \mid & \text{(contradiction concept)} \\
& \neg C \mid & \text{(complement)} \\
& C \sqcap C \mid & \text{(conjunction)} \\
& C \sqcup C \mid & \text{(disjunction)} \\
& \forall R.C \mid & \text{(value restriction)} \\
& \exists R.C \mid & \text{(existential restriction)}
\end{array}
$$

where A ranges over atomic concepts, R over atomic roles, and C over complex concepts. Recalling our running example, the statements $\mathsf{Foreigner} \sqcap \mathsf{EUcitizen}$, $\exists\mathsf{passport}.\top$, $\forall\mathsf{passport}.\mathsf{EU}$, and $\forall\mathsf{passport}.\neg\mathsf{EU}$ are complex concepts in $\mathcal{ALC}$.

We use individuals to describe a specific state of affairs in terms of concepts and roles. We use $a, b, \ldots$ to denote individuals. In our example, JAN and POLAND are individuals of which we can assert, respectively, the properties $\mathsf{EUcitizen}$ and EU. The intended meaning of such assertions is that JAN has EU citizenship and POLAND is a member of the European community. Individuals and assertions about them allow us to give a description of the world.

Definition 1. *An* interpretation $\mathcal{I}$ *is a tuple* $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ *such that* $\Delta^{\mathcal{I}}$ *is a nonempty set and* $\cdot^{\mathcal{I}}$ *a function mapping:*

- *every concept to a subset of* $\Delta^{\mathcal{I}}$
- *every role to a subset of* $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
- *every individual to an element of* $\Delta^{\mathcal{I}}$

Given an interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, $\Delta^{\mathcal{I}}$ is the interpretation *domain*, and $\cdot^{\mathcal{I}}$ the associated interpretation function. If a is an individual name, A an atomic concept, R an atomic role, and C, D concepts, we have:

$$\begin{aligned}
a^{\mathcal{I}} &\in \Delta^{\mathcal{I}} \\
A^{\mathcal{I}} &\subseteq \Delta^{\mathcal{I}} \\
R^{\mathcal{I}} &\subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \\
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} &= \emptyset \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \ : \ \forall b.(a,b) \in R^{\mathcal{I}} \text{ implies } b \in C^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \ : \ \exists b.(a,b) \in R^{\mathcal{I}} \text{ and } b \in C^{\mathcal{I}}\}
\end{aligned}$$

In $\mathcal{ALC}$ we also have *terminological axioms* (axioms, for short). These are statements of the form $C \equiv D$ and $C \sqsubseteq D$. Axioms of the first kind are called *concept definitions* (alias *equalities*). Those of the second kind are called *concept inclusion axioms* (alias *inclusions* or *subsumptions*). If C and D are both complex concepts, then $C \sqsubseteq D$ is called a *general concept inclusion* axiom (GCI).

An interpretation $\mathcal{I}$ satisfies a concept definition $C \equiv D$ (noted $\models^{\mathcal{I}} C \equiv D$) if $C^{\mathcal{I}} = D^{\mathcal{I}}$. Intuitively, $C \equiv D$ establishes a definition for concept C in terms of D. In our example, we have $\mathsf{DoubleCitizen} \equiv \mathsf{Foreigner} \sqcap \mathsf{EUcitizen}$, which gives both necessary and sufficient conditions to be a person with double citizenship.

An interpretation $\mathcal{I}$ satisfies a subsumption $C \sqsubseteq D$ (noted $\models^{\mathcal{I}} C \sqsubseteq D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Intuitively, $C \sqsubseteq D$ means that concept C is more specific than concept D. In our example we have $\mathsf{DoubleCitizen} \sqsubseteq \mathsf{EUcitizen}$, which says that a person with double citizenship is a specialization of a European citizen. We also have $\mathsf{Passenger} \sqsubseteq \exists\mathsf{passport}.\top$, saying that a necessary condition to be a passenger is having a passport. Concept inclusion axioms are used when one is not able to completely define a concept: in the last example, a passenger may have many other properties of which the knowledge engineer was not necessarily aware when modeling the description.

We call a (finite) set of terminological axioms a *terminology*, alias TBox. We denote TBoxes by $\mathcal{T}$. An interpretation $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ (noted $\models^{\mathcal{I}} \mathcal{T}$) if $\models^{\mathcal{I}} C \sqsubseteq D$ for all $C \sqsubseteq D \in \mathcal{T}$. An axiom $C \sqsubseteq D$ is a *consequence* of a TBox $\mathcal{T}$ (noted $\mathcal{T} \models C \sqsubseteq D$) if for every interpretation $\mathcal{I}$, $\models^{\mathcal{I}} \mathcal{T}$ implies $\models^{\mathcal{I}} C \sqsubseteq D$.

Henceforth we can suppose w.l.o.g. that TBoxes are *linearized*, i.e., $\mathcal{T}$ only contains inclusion axioms (no concept definitions), and see $C \equiv D$ as just as an abbreviation for $C \sqsubseteq D$ and $D \sqsubseteq C$.

A *concept assertion* is a statement about an individual with respect to some concept. We denote by $C(a)$ the fact that a belongs to (the interpretation of) concept C. In our example, the assertion $\mathsf{Foreigner}(\mathsf{JOHN})$ says that JOHN is a non-European citizen, and that all properties a foreigner has (e.g. possessing a non-EU passport) apply to JOHN as well.

A *role assertion* establishes a relationship between two individuals. If a, b are individuals and R is a role name, then $R(a, b)$ asserts that b is a *filler* of the role R for a. In our example, the role assertion $\mathsf{refund}(\mathsf{JOHN}, \mathsf{VAT})$ states that JOHN can claim the refund of the value added tax when leaving the airport.

An interpretation $\mathcal{I}$ satisfies a concept assertion $C(a)$ (noted $\models^{\mathcal{I}} C(a)$) if $a^{\mathcal{I}} \in C^{\mathcal{I}}$, and a role assertion $R(a, b)$ (noted $\models^{\mathcal{I}} R(a, b)$) if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$.

A (finite) set of concept and role assertions define an ABox. We denote ABoxes by $\mathcal{A}$. An interpretation $\mathcal{I}$ is a *model* of an ABox $\mathcal{A}$ (noted $\models^{\mathcal{I}} \mathcal{A}$) if $\mathcal{I}$ satisfies every assertion in $\mathcal{A}$. A concept assertion $C(a)$ (resp. a role assertion $R(a, b)$) is a *consequence* of an ABox $\mathcal{A}$, noted $\mathcal{A} \models C(a)$ (resp. $\mathcal{A} \models R(a, b)$), if for every interpretation $\mathcal{I}$, $\models^{\mathcal{I}} \mathcal{A}$ implies $\models^{\mathcal{I}} C(a)$ (resp. $\models^{\mathcal{I}} R(a, b)$).

A *knowledge base* is a tuple $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ is a TBox and $\mathcal{A}$ an ABox. An interpretation $\mathcal{I}$ is a model of $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$ if $\models^{\mathcal{I}} \mathcal{T}$ and $\models^{\mathcal{I}} \mathcal{A}$. Logical consequence of an axiom $C \sqsubseteq D$, of a concept assertion $C(a)$ and of a role assertion $R(a, b)$ from a knowledge base Σ is defined in the standard way.

In the rest of this paper we are going to restrict ourselves only to the TBox component of knowledge bases.

3 Role-Based Decomposition

Here we give a novel way of decomposing ontologies. Let $\mathfrak{Roles} = \{R_1, R_2, \ldots\}$ be the set of all role names of a domain. Let $\mathit{roles}(C \sqsubseteq D)$ return the set of role names occurring in an axiom $C \sqsubseteq D$. For instance $\mathit{roles}(C \equiv \exists R_1.D \sqcap \forall R_2.E) = \{R_1, R_2\}$. Moreover, for a TBox $\mathcal{T}$, let $\mathit{roles}(\mathcal{T}) = \bigcup_{C \sqsubseteq D \in \mathcal{T}} \mathit{roles}(C \sqsubseteq D)$.

With that we define a role-based classification of axioms.

Definition 2. *A* boolean axiom *is an axiom $C \sqsubseteq D$ such that $\mathit{roles}(C \sqsubseteq D) = \emptyset$. If $\mathit{roles}(C \sqsubseteq D) \neq \emptyset$, $C \sqsubseteq D$ is a* non-boolean axiom.

If $\mathscr{R} \subseteq \mathfrak{Roles}$, $\mathscr{R} \neq \emptyset$, then we define

$$\mathcal{T}^{\mathscr{R}} = \{C \sqsubseteq D \in \mathcal{T} \ : \ \mathit{roles}(C \sqsubseteq D) \cap \mathscr{R} \neq \emptyset\}$$

Hence, $\mathcal{T}^{\mathscr{R}}$ contains all non-boolean axioms of the terminology $\mathcal{T}$ whose roles appear in $\mathscr{R}$. For $\mathscr{R} = \emptyset$, $\mathcal{T}^{\emptyset} = \{C \sqsubseteq D \in \mathcal{T} \ : \ \mathit{roles}(C \sqsubseteq D) = \emptyset\}$ is the set of all boolean axioms of a knowledge base.

For example, if

$$\mathcal{T} = \left\{ \begin{array}{c} \mathsf{Passenger} \sqsubseteq \exists \mathsf{passport}.\top,\ \mathsf{EUcitizen} \equiv \forall \mathsf{passport}.\mathsf{EU}, \\ \mathsf{Foreigner} \equiv \forall \mathsf{passport}.\neg \mathsf{EU},\ \mathsf{Foreigner} \sqsubseteq \exists \mathsf{refund}.\mathsf{Tax}, \\ \mathsf{DoubleCitizen} \equiv \mathsf{Foreigner} \sqcap \mathsf{EUcitizen} \end{array} \right\}$$

then we have

$$\mathcal{T}^{\{\mathsf{refund}\}} = \{\mathsf{Foreigner} \sqsubseteq \exists \mathsf{refund}.\mathsf{Tax}\}$$

and

$$\mathcal{T}^{\emptyset} = \{\mathsf{DoubleCitizen} \equiv \mathsf{Foreigner} \sqcap \mathsf{EUcitizen}\}$$

For parsimony's sake, we write $\mathcal{T}^R$ instead of $\mathcal{T}^{\{R\}}$.

Given these fundamental concepts, we are able to formally define modularity for ontologies in description logics.

4 Modular TBoxes

We can suppose from now on that $\mathcal{T}$ is *partitioned*, in the sense that $\{\mathcal{T}^{\emptyset}\} \cup \{\mathcal{T}^{R_i} : R_i \in \mathfrak{Roles}\}$ is a partition[1] of $\mathcal{T}$. We thus exclude $\mathcal{T}^{R_i}$ containing more than one role name, which means that complex concepts with nested roles are not allowed. We thus make it a hypothesis:

$$\{\mathcal{T}^{\emptyset}\} \cup \{\mathcal{T}^{R_i} : R_i \in \mathfrak{Roles}\} \text{ partitions } \mathcal{T} \tag{H}$$

We are interested in the following principle of modularity:

Definition 3. *A terminology $\mathcal{T}$ is* modular *if and only if for every $C \sqsubseteq D$,*

$$\mathcal{T} \models C \sqsubseteq D \textit{ implies } \mathcal{T}^{roles(C \sqsubseteq D)} \cup \mathcal{T}^{\emptyset} \models C \sqsubseteq D.$$

Modularity means that when investigating whether $C \sqsubseteq D$ is a consequence of $\mathcal{T}$, the only axioms in $\mathcal{T}$ that are relevant are those whose role names occur in $C \sqsubseteq D$ and the boolean axioms in $\mathcal{T}^{\emptyset}$.

This is reminiscent of interpolation [4], which for the case of roles says:

Definition 4. *A terminology $\mathcal{T}$ has the* interpolation property *if and only if for every axiom $C \sqsubseteq D$, if $\mathcal{T} \models C \sqsubseteq D$, then there is a terminology $\mathcal{T}_{C \sqsubseteq D}$ such that*

- *$roles(\mathcal{T}_{C \sqsubseteq D}) \subseteq roles(\mathcal{T}) \cap roles(C \sqsubseteq D)$*
- *$\mathcal{T} \models C' \sqsubseteq D'$ for every $C' \sqsubseteq D' \in \mathcal{T}_{C \sqsubseteq D}$*
- *$\mathcal{T}_{C \sqsubseteq D} \models C \sqsubseteq D$*

[1] Remembering, $\{\mathcal{T}^{\emptyset}\} \cup \{\mathcal{T}^{R_i} : R_i \in \mathfrak{Roles}\}$ partitions $\mathcal{T}$ if and only if $\mathcal{T} = \mathcal{T}^{\emptyset} \cup \bigcup_{R_i \in \mathfrak{Roles}} \mathcal{T}^{R_i}$, and $\mathcal{T}^{\emptyset} \cap \mathcal{T}^{R_i} = \emptyset$, and $\mathcal{T}^{R_i} \cap \mathcal{T}^{R_j} = \emptyset$, if $i \neq j$. Note that $\mathcal{T}^{\emptyset}$ and $\mathcal{T}^{R_i}$ might be empty.

Our definition of modularity is a strengthening of interpolation because it requires $\mathcal{T}_{C \sqsubseteq D}$ to be a subset of $\mathcal{T}$.

Contrary to interpolation however, modularity does not generally hold. Clearly if the Hypothesis (H) is not satisfied, then modularity fails. To witness, consider

$$\mathcal{T} = \{C \equiv \forall R_1.\forall R_2.C', \forall R_1.\forall R_2.C' \equiv D\}$$

Then $\mathcal{T} \models C \equiv D$, but $\mathcal{T}^{\emptyset} \not\models C \equiv D$.

Nevertheless even under our hypothesis modularity may fail to hold. For example, let

$$\mathcal{T} = \{C \sqcup \forall R.\bot \equiv \top, C \sqcup \exists R.\top \equiv \top\}$$

Then $\mathcal{T}^{\emptyset} = \emptyset$, and $\mathcal{T}^{R} = \mathcal{T}$. Now $\mathcal{T} \models C$, but clearly $\mathcal{T}^{\emptyset} \not\models C$.

How can we know whether a given TBox $\mathcal{T}$ is modular? The following criterion is simpler:

Definition 5. *A terminology $\mathcal{T}$ is* boolean-modular *if and only if for every boolean axiom $C \sqsubseteq D$,*

$$\mathcal{T} \models C \sqsubseteq D \textit{ implies } \mathcal{T}^{\emptyset} \models C \sqsubseteq D.$$

With that we guarantee modularity:

Theorem 1 ([12]). *Let $\mathcal{T}$ be a partitioned terminology. If $\mathcal{T}$ is boolean-modular, then $\mathcal{T}$ is modular.*

In the rest of the paper we investigate how it can be automatically checked whether a given terminology $\mathcal{T}$ is modular and how to make it modular, if needed. We do this for a version of $\mathcal{ALC}$ with a restriction on the form of the axioms we can state in a TBox.

5 Soundness and Completeness for a Fragment of $\mathcal{ALC}$

Definition 6. *A concept C is a* boolean concept *if $roles(C) = \emptyset$.*

We here make a syntactical restriction on the form of non-boolean axioms in our TBoxes.

Definition 7. *If C is a boolean concept, then $\forall R.C$ is a* boolean value restriction*, and $\exists R.C$ is a* boolean existential restriction*.*

In this section we suppose that:

All value/existential restrictions in a knowledge base are boolean value/existential restrictions. (H2)

Our fragment differs from $\mathcal{ALC}$ just in the sense that only boolean concepts are allowed in the scope of a quantification over a role. We observe however that we could allow for axioms with nested roles like $C \equiv \forall R_1.\forall R_2.D$ and GCIs like $\forall R_3.E \sqsubseteq \forall R_4.F$. For that it would suffice to adapt an existing technique of *subformula renaming* [17] in the literature on classical logic [14, 2, 3] to recursively replace complex concepts with some new concepts, stating definitions for these as global axioms. For instance, $C \equiv \forall R_1.\forall R_2.D$ should then be rewritten as $C \equiv \forall R_1.C'$ and $C' \equiv \forall R_2.D$, and $\forall R_3.E \sqsubseteq \forall R_4.F$ could be replaced by $E' \sqsubseteq \forall R_4.F$ and $E' \equiv \forall R_4.E$, where C', E' are new concept names. It is known that subformula renaming is satisfiability preserving and can be computed in polynomial time [13]. However it remains to assess the impact the introduction of new concept names can have on the intuition about the original ontology.

Our central hypothesis here is that the different types of axioms in a given terminology should be neatly separated and only interfere in one sense: boolean axioms together with non-boolean axioms for role R may have consequences that do not follow from the non-boolean axioms for R alone. The other way round, non-boolean axioms should not allow to infer new boolean axioms. That is what we expect modularity of TBoxes to establish and we develop it in the sequel.

Definition 8. *A boolean inclusion axiom* $C \sqsubseteq D$ *is an* implicit boolean inclusion axiom *of a terminology* $\mathcal{T}$ *if and only if* $\mathcal{T} \models C \sqsubseteq D$ *and* $\mathcal{T}^{\emptyset} \not\models C \sqsubseteq D$.

In our running example, DoubleCitizen $\sqsubseteq \neg$Passenger is an example of an implicit boolean inclusion axiom.

With Algorithm 1 below we can check whether a TBox has such implicit axioms. The idea is as follows: for each pair of axioms $C \sqsubseteq \exists R.D$ and $E \sqsubseteq \forall R.F$ in $\mathcal{T}$ such that F conflicts with D, i.e., $\mathcal{T} \models D \sqcap F \sqsubseteq \bot$, if $\mathcal{T}^{\emptyset} \cup \{C \sqcap E\}$ is satisfiable and $\mathcal{T}^{\emptyset} \not\models C \sqsubseteq \neg E$, mark $C \sqsubseteq \neg E$ as an implicit boolean inclusion axiom.

Algorithm 1. Deciding existence of implicit boolean inclusion axioms

input: a TBox $\mathcal{T}$
output: a set of implicit boolean inclusion axioms $\mathcal{T}^{\emptyset}_{imp}$

$\mathcal{T}^{\emptyset}_{imp} := \emptyset$
for all $R \in \mathfrak{Roles}$ **do**
  for all $\{C_1 \sqsubseteq \exists R.D_1, \ldots, C_n \sqsubseteq \exists R.D_n\} \subseteq \mathcal{T}$ **do**
    for all $\{E_1 \sqsubseteq \forall R.F_1, \ldots, E_m \sqsubseteq \forall R.F_m\} \subseteq \mathcal{T}$ **do**
      if $\mathcal{T}^{\emptyset} \not\models \bigsqcap_{1 \leq i \leq n} C_i \sqcap \bigsqcap_{1 \leq j \leq m} E_i \sqsubseteq \bot$ **and**
        $\mathcal{T}^{\emptyset} \models \bigsqcap_{1 \leq i \leq n} D_i \sqcap \bigsqcap_{1 \leq j \leq m} F_i \sqsubseteq \bot$ **then**
        $\mathcal{T}^{\emptyset}_{imp} := \mathcal{T}^{\emptyset}_{imp} \cup \{\bigsqcap_{1 \leq i \leq n} C_i \sqsubseteq \bigsqcup_{1 \leq j \leq m} \neg E_i\}$

Theorem 2. *Algorithm 1 terminates.*

Proof. Straightforward from finiteness of $\mathcal{T}$.

Lemma 1. *Let $\mathcal{T}^{\emptyset}_{imp^*}$ be the output of Algorithm 2 on input $\mathcal{T}$. Then every $C \sqsubseteq D \in \mathcal{T}^{\emptyset}_{imp}$ is an implicit boolean inclusion axiom of $\mathcal{T}$.*

Converse of Lemma 1 does not hold. Indeed, consider the quite simple TBox:

$$\mathcal{T} = \left\{ \begin{array}{c} C_n \sqsubseteq \bot, \\ C_{i-1} \sqsubseteq \forall R.C_i,\ 1 \leq i \leq n, \\ \top \sqsubseteq \exists R.\top \end{array} \right\}$$

Thus, $\mathcal{T} \models C_i \sqsubseteq \bot$, for $0 \leq i \leq n$, but running Algorithm 1 returns only $\mathcal{T}^{\emptyset}_{imp} = \{C_{n-1} \sqsubseteq \bot\}$. This suggests that it is necessary to iterate the algorithm in order to find all implicit boolean inclusion axioms. Before doing that we observe that:

Theorem 3. *A terminology $\mathcal{T}$ is modular if and only if $\mathcal{T}^{\emptyset}_{imp} = \emptyset$.*

Considering the example just above, we can see that running Algorithm 1 on $\mathcal{T} \cup \{C_{n-1} \sqsubseteq \bot\}$ will give us $\mathcal{T}^{\emptyset}_{imp} = \{C_{n-2} \sqsubseteq \bot\}$. This means that some of the implicit boolean inclusion axioms of a terminology may be needed in order to derive others. Hence, Algorithm 1 must be iterated to get $\mathcal{T}$ modular. This is achieved with the following algorithm, which iteratively feeds the set of boolean axioms considered into the **if**-test of Algorithm 1:

Algorithm 2. Finding all implicit boolean inclusion axioms

---

input: a TBox $\mathcal{T}$
output: $\mathcal{T}^{\emptyset}_{imp^*}$, the set of all implicit boolean inclusion axioms of $\mathcal{T}$

```
T^∅_imp* := ∅
repeat
   T^∅_imp := find_imp_bia(T ∪ T^∅_imp*) {a call to Algorithm 1}
   T^∅_imp* := T^∅_imp* ∪ T^∅_imp
until T^∅_imp = ∅
```

Theorem 4. *Algorithm 2 terminates.*

Theorem 5. *Let $\mathcal{T}^{\emptyset}_{imp^*}$ be the output of Algorithm 2 on input $\mathcal{T}$. Then*

1. *$\mathcal{T} \cup \{\mathcal{T}^{\emptyset}_{imp^*}\}$ is modular;*
2. *$\mathcal{T} \models \bigsqcap\{\mathcal{T}^{\emptyset}_{imp^*}\}$.*

Corollary 1. *For all boolean inclusion axioms $C \sqsubseteq D$, $\mathcal{T} \models C \sqsubseteq D$ if and only if $\mathcal{T} \cup \{\mathcal{T}^{\emptyset}_{imp^*}\} \models C \sqsubseteq D$.*

This establishes that Algorithm 2 finds all implicit boolean inclusion axioms of a given terminology $\mathcal{T}$. Hence, adding such axioms to the original set of boolean axioms $\mathcal{T}^{\emptyset}$ guarantees modularity of $\mathcal{T}$.

We want to point out, however, that the algorithm only catches implicit boolean inclusion axioms. Deciding whether they are intuitive remains the knowledge engineer's task, and only she can carry out changes in the knowledge base in order to accommodate them in or discard them from the description. In our running example, the inclusion $\mathsf{DoubleCitizen} \sqsubseteq \neg\mathsf{Passenger}$ is not intuitive and should then be contracted from the terminology.

Algorithms 1 and 2 are generalizations/extensions of the method for PDL given in [12] where (in terms of description logics) only existential restrictions of the form $C \sqsubseteq \exists R.\top$ were allowed.

6 The Role of Modularity in Reasoning Services

The following result is important in the ontology building phase:

Theorem 6. *Let $\mathcal{T}$ and $C \sqsubseteq D$ be such that $\mathcal{T} \not\models \top \sqsubseteq \bot$. If $\mathcal{T}$ is modular, then $\mathcal{T} \cup \{C \sqsubseteq D\} \models \top \sqsubseteq \bot$ if and only if $\mathcal{T}^{\emptyset} \cup \mathcal{T}^{roles(C \sqsubseteq D)} \cup \{C \sqsubseteq D\} \models \top \sqsubseteq \bot$.*

This theorem says that under modularity consistency of a new learned axiom $C \sqsubseteq D$ w.r.t. a consistent TBox reduces to consistency check of the axioms that are relevant to $C \sqsubseteq D$.

Theorem 7. *If $\mathcal{T}$ is modular, then $\mathcal{T} \models \top \sqsubseteq \bot$ if and only if $\mathcal{T}^{\emptyset} \models \top \sqsubseteq \bot$.*

Hence, if there are no implicit boolean inclusion axioms, then consistency of the whole terminology can be checked by just checking consistency of $\mathcal{T}^{\emptyset}$.

It turns out that checking whether a concept C is the *least common subsumer* (*lcs*) of a set of concepts, i.e., the minimal concept that subsumes all other concepts in question [1], is also optimized under modularity:

Theorem 8. *Let Γ be a set of concepts. If $\mathcal{T}$ is modular, then C is the lcs of Γ w.r.t. $\mathcal{T}$ if and only if C is the lcs of Γ w.r.t. $\mathcal{T}^{\emptyset} \cup \mathcal{T}^{roles(C)}$.*

For $\mathcal{T}$ a TBox, we define $\mathcal{T}_{\forall}^{R} = \{C \sqsubseteq \forall R.D \;:\; C \sqsubseteq \forall R.D \in \mathcal{T}\}$, i.e., $\mathcal{T}_{\forall}^{R}$ contains all non-boolean axioms in the TBox $\mathcal{T}$ with value restrictions for role R.

Theorem 9. *If $\mathcal{T}$ is modular, then*

$$\mathcal{T} \models C \sqsubseteq \forall R.D \text{ if and only if } \mathcal{T}^{\emptyset} \cup \mathcal{T}_{\forall}^{R} \models C \sqsubseteq \forall R.D.$$

This means that under our modularity principle we have modularity inside the module for non-boolean axioms, too: when deducing an axiom with value restrictions we do not need to consider axioms with existential restrictions.

The existential restriction counterpart of Theorem 9, however, does not hold. To witness, from the modular description $\{\forall R.C \sqcup D, \exists R.\neg C\}$ we conclude $\exists R.D$, but $\{\exists R.\neg C\} \not\models \exists R.D$. Nevertheless, we can establish a result if only the universal concept ($\top$) is allowed in the scope of existential restrictions. For that we define $\mathcal{T}_{\exists}^{R} = \{C \sqsubseteq \exists R.\top \;:\; C \sqsubseteq \exists R.\top \in \mathcal{T}\}$.

Theorem 10. *If $\mathcal{T}$ is modular, then*

$$\mathcal{T} \models C \sqsubseteq \exists R.\top \text{ if and only if } \mathcal{T}^{\emptyset} \cup \mathcal{T}_{\exists}^{R} \models C \sqsubseteq \exists R.\top.$$

Let $\mathcal{T}_{\forall}^{R_1,\ldots,R_n} = \bigcup_{1 \leq i \leq n} \mathcal{T}_{\forall}^{R_i}$. The following theorem shows that under modularity deduction of an axiom based on nested value restrictions does not need the axioms based on existential restrictions:

Theorem 11. *If $\mathcal{T}$ is modular, then $\mathcal{T} \models C \sqsubseteq \forall R_1 \ldots \forall R_n.D$ if and only if $\mathcal{T}^{\emptyset} \cup \mathcal{T}_{\forall}^{R_1,\ldots,R_n} \models C \sqsubseteq \forall R.D$.*

The same result holds for deductions of axioms based on existential restrictions under the assumption that only $\top$ is allowed in the scope of $\exists$. Let $\mathcal{T}_{\exists}^{R_1,\ldots,R_n} = \bigcup_{1 \leq i \leq n} \mathcal{T}_{\exists}^{R_i}$.

Theorem 12. *If $\mathcal{T}$ is modular, then $\mathcal{T} \models C \sqsubseteq \exists R_1 \ldots \exists R_n.\top$ if and only if $\mathcal{T}^{\emptyset} \cup \mathcal{T}_{\exists}^{R_1,\ldots,R_n} \models C \sqsubseteq \exists R.\top$.*

7 Concluding Remarks

We defined here a modularity paradigm for ontologies in description logics and pointed out some of the problems that arise if it is not satisfied, even if the ontology is consistent. In particular we have argued that the boolean part of a description could influence but should not be influenced by the role-based one.

We have seen that the presence of implicit boolean inclusion axioms is a sign that we possibly have slipped up in designing the ontology in question. We showed how to detect this problem in a fragment of $\mathcal{ALC}$ with a syntactical restriction on its formulas. With Algorithm 2 we have a sound and complete decision procedure for such a task. Moreover, the output of the algorithm gives us guidelines that can help correcting the ontology.

We could also use full $\mathcal{ALC}$, in this case our method is sound but not complete. As an example, let $\mathcal{T} = \{C \equiv \forall R_1.\forall R_2.D, C' \equiv \forall R_1.\exists R_2.\neg D, \top \equiv \exists R_1.\top\}$. We have $\mathcal{T} \models C \sqsubseteq \neg C'$, but running Algorithm 2 on $\mathcal{T}$ gives $\mathcal{T}_{imp*}^{\emptyset} = \emptyset$.

It could be argued that unintuitive consequences in ontologies are mainly due to badly written axioms and not to lack of modularity. True enough, but what we presented here is the case that making an ontology modular gives us a tool to detect some of such problems and correct it. (But note that we do not claim to correct badly written axioms automatically and once for all.) Besides this, having separate entities in the ontology and controlling their interaction help us to localize where the problems are, which is crucial for real world applications.

As our theorems show (proofs were omitted due to lack of space), being modular is a useful feature of terminologies w.r.t. reasoning: beyond being a reasonable principle of design that helps structuring data, it clearly restricts the search space, and thus makes reasoning easier.

The first work on formalizing modularity in logical systems in general seems to be due to Garson [6]. Modularity of theories in reasoning about actions was originally defined in [10] and extensively developed in [12, 9]. A different viewpoint of that can be found in [11], where modularity of action theories is assessed from a more software engineering oriented perspective. The present work has been inspired by ideas in the referred approaches. Following [6], a modularization technique for ontologies in DL different from ours is addressed in [5].

Our notion of modularity is related to uniform interpolation for TBoxes [7]. Let *concepts*($\mathcal{T}$) denote the concept names occurring in a TBox $\mathcal{T}$. Given $\mathcal{T}$ and a signature $\mathcal{S} \subseteq$ *concepts*($\mathcal{T}$) $\cup$ *roles*($\mathcal{T}$), a TBox $\mathcal{T}^{\mathcal{S}}$ over (*concepts*($\mathcal{T}$) $\cup$ *roles*($\mathcal{T}$)) $\setminus \mathcal{S}$ is a *uniform interpolant* of $\mathcal{T}$ outside $\mathcal{S}$ if and only if:

- $\mathcal{T} \models \mathcal{T}^{\mathcal{S}}$;
- $\mathcal{T}^{\mathcal{S}} \models C \sqsubseteq D$ for every $C \sqsubseteq D$ that has no occurrences of symbols from $\mathcal{S}$.

It is not difficult to see that a partition $\{\mathcal{T}^{\emptyset}\} \cup \{\mathcal{T}^{R_i} \ : \ R_i \in \mathfrak{Roles}\}$ is modular if and only if every $\mathcal{T}^{R_i}$ is a uniform interpolant of $\mathcal{T}$ outside *roles*($\mathcal{T}$) $\setminus \{R_i\}$. In [16] there are complexity results for computing uniform interpolants in $\mathcal{ALC}$.

In [7] a notion of conservative extension is defined that is similar to our modularity. There, $\mathcal{T}_1 \cup \mathcal{T}_2$ is a *conservative extension* of $\mathcal{T}_1$ if and only if for all concepts C, D built from *concepts*($\mathcal{T}_1$) $\cup$ *roles*($\mathcal{T}_1$), $\mathcal{T}_1 \cup \mathcal{T}_2 \models C \sqsubseteq D$ implies $\mathcal{T}_1 \models C \sqsubseteq D$.

Given our Theorem 1, we can show that checking for modularity can be reduced to checking for conservative extensions of $\mathcal{T}^{\emptyset}$. Indeed, supposing that the signature of $\mathcal{T}^{\emptyset}$ is the set of all concept names, we have that $\mathcal{T}$ is modular if and only if for every role R_i, $\mathcal{T}^{R_i} \cup \mathcal{T}^{\emptyset}$ is a conservative extension of $\mathcal{T}^{\emptyset}$.

We plan to pursue further work on extensions of our method to more expressive description logics. Another extension that we foresee is generalizing modularity to also take into account ABoxes. In this case our algorithms should be adapted so that implicit interactions between terminologies and assertions can be caught.

Because interactions between TBoxes and ABoxes may lead to inconsistency, ontology update and revision should be considered, too. We are currently investigating update of terminologies based on the method given in [8], for which satisfaction of modularity shows to be fruitful.

Acknowledgments

We are grateful to the anonymous referees for useful comments on an earlier version of this paper. Thanks to Bernardo Cuenca Grau for useful and interesting discussions about the topic of this work. We also would like to thank Meghyn Bienvenu for comments about DL terminology and notation.

Ivan Varzinczak has been supported by a fellowship from the government of the FEDERATIVE REPUBLIC OF BRAZIL. Grant: CAPES BEX 1389/01-7.

References

1. F. Baader and W. Nutt. Basic description logics. In F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors, *Description Logic Handbook*, chapter 2, pages 47–100. Cambridge University Press, 2003.

2. T. Boy de la Tour. Minimizing the number of clauses by renaming. In M.E. Stickel, editor, *Proc. 10th International Conference on Automated Deduction (CADE'90)*, pages 558–572, London, 1990. Springer-Verlag. LNCS, vol. 449.
3. T. Boy de la Tour. An optimality result for clause form translation. *J. of Symbolic Computation*, 14(4):283–301, 1992.
4. W. Craig. Linear reasoning. A new form of the Herbrand-Gentzen theorem. *J. of Symbolic Logic*, 22:250–268, 1957.
5. B. Cuenca Grau, B. Parsia, E. Sirin, and A. Kalyanpur. Modularity and web ontologies. In *Proc. 10th Intl. Conf. on Knowledge Representation and Reasoning (KR'2006)*, pages 198–208, Lake District, 2006. Morgan Kaufmann Publishers.
6. J. Garson. Modularity and relevant logic. *Notre Dame J. of Formal Logic*, 30(2):207–223, 1989.
7. S. Ghilardi, C. Lutz, and F. Wolter. Did I damage my ontology? a case for conservative extensions in description logic. In *Proc. 10th Intl. Conf. on Knowledge Representation and Reasoning (KR'2006)*, pages 187–197, Lake District, 2006. Morgan Kaufmann Publishers.
8. A. Herzig, L. Perrussel, and I. Varzinczak. Elaborating domain descriptions. In *Proc. 17th Eur. Conf. on Artificial Intelligence (ECAI'06)*, Riva del Garda, 2006. IOS Press.
9. A. Herzig and I. Varzinczak. Metatheory of actions: beyond consistency. To appear.
10. A. Herzig and I. Varzinczak. Domain descriptions should be modular. In R. López de Mántaras and L. Saitta, editors, *Proc. 16th Eur. Conf. on Artificial Intelligence (ECAI'04)*, pages 348–352, Valencia, 2004. IOS Press.
11. A. Herzig and I. Varzinczak. Cohesion, coupling and the meta-theory of actions. In L. Kaelbling and A. Saffiotti, editors, *Proc. 19th Intl. Joint Conf. on Artificial Intelligence (IJCAI'05)*, pages 442–447, Edinburgh, 2005. Morgan Kaufmann Publishers.
12. A. Herzig and I. Varzinczak. On the modularity of theories. In R. Schmidt, I. Pratt-Hartmann, M. Reynolds, and H. Wansing, editors, *Advances in Modal Logic*, volume 5, pages 93–109. King's College Publications, 2005. Selected papers of AiML 2004 (also available at `http://www.aiml.net/volumes/volume5`).
13. A. Nonnengart and C. Weidenbach. Computing small clause normal forms. In A. Robinson and A. Voronkov, editors, *Handbook of automated reasoning*, pages 335–367. 2001.
14. D. Plaisted and S. Greenbaum. A structure-preserving clause form translation. *J. of Symbolic Computation*, 2(3):293–304, 1986.
15. H. Stuckenschmidt and M. Klein. Integrity and change in modular ontologies. In V. Sorge, S. Colton, M. Fisher, and J. Gow, editors, *Proc. 18th Intl. Joint Conf. on Artificial Intelligence (IJCAI'03)*, pages 900–908, Acapulco, 2003. Morgan Kaufmann Publishers.
16. B. ten Cat, W. Conradie, M. Marx, and Y. Venema. Definitorially complete description logics. In *Proc. 10th Intl. Conf. on Knowledge Representation and Reasoning (KR'2006)*, pages 79–89, Lake District, 2006. Morgan Kaufmann Publishers.
17. G.S. Tseitin. On the complexity of derivation in propositional calculus. In A. Slisenko, ed., Studies in Constructive Mathematics and Mathematical Logics, Part II, 1968.