The Effect of Modification and Update Propagation on Modular Ontologies

Faezeh Ensan, Ebrahim Bagheri and Weichang Du

Faculty of Computer Science,
University of New Brunswick, Fredericton, Canada
{faezeh.ensan, ebagheri, wdu}@unb.ca

Abstract. One of the most important issues in the development of ontologies is dealing with revisions and updates, which becomes even more prominent in modular representations of ontologies, in light of the fact that an update in one ontology module may lead to unintended consequences in other ontology modules due to their coupling. In this paper, we provide a basis for updating modular ontologies and specially those that are represented by the interface-based formalism for modular ontologies. We illustrate different scenarios where modifications and propagation of updates in connected ontologies induce inconsistencies. An algorithm for handling these inconsistencies is provided that updates modular ontologies with TBox modifications.

1 Introduction

Ontologies are the main building blocks of the Semantic Web that provide a basis for formally defining the concepts of a domain of discourse and their relationships. Recently, a considerable amount of research has been dedicated to the development of ontologies in a modular manner [16]. Distributed Description Logics [3], E-connections [22] and PDL [1] are among the formalisms that provide foundations for modular ontologies.

In [10, 8], we have introduced the Interface-Based Formalism for modular ontologies (IBF). Based on this formalism, a modular ontology can be defined as a set of self-contained ontology modules, which may present some perspectives of their knowledge bases through various interfaces. The main feature of IBF is its support for knowledge encapsulation, i.e., it allows ontologies to define their main content using well-defined interfaces, such that their knowledge base can only be accessed by other ontologies through these interfaces. An important implication of this formalism is that ontology modules can be developed completely independent of each others signature and language. Such modules are free to utilize the required knowledge segments of the others. The interface-based formalism enjoys a great expressiveness power, which allows an ontology module to create its knowledge base from the other modules’ knowledge expressed through their interfaces.

One of the important issues in the development of ontologies is dealing with revisions and updates [12]. The problem is that a modification in a knowledge
base may give rise to inconsistencies that should be resolved properly. In the community of belief revision, different principles and postulates have been proposed for updating belief bases [15, 21]. The principle of minimal change is among the most important ones that states that as much information as possible must be preserved for resolving consequential inconsistencies [21].

Few proposals exist in the literature that recast belief revision principles and postulates for ontologies and description logics [13]. There are also some other works that analyze the revision of TBox and ABox of ontologies. The authors in [17] provide a survey of some available approaches for dealing with inconsistencies in changing ontologies and compare them. In [5], the update of ABox in description logic knowledge bases is analyzed and an algorithm for revising DL-Lite [4] knowledge bases is provided. The authors in [24] propose two revision operators for description logics when the consequential inconsistencies that are arisen by revisions are due to objects explicitly introduced in the Abox. They propose a method for weakening the knowledge base in order to satisfy the principle of minimal change. In [23], the weakening approach of [2] is exploited to perform inconsistency resolution in stratified knowledge bases. This work can be seen as relevant to the update and revision of description logic knowledge bases if we consider higher priority knowledge base as a modification on those with lower priorities.

There are also some work in the literature that attempt to debug inconsistent terminologies and knowledge bases. Such stream of research can be exploited for resolving inconsistencies that are caused by updates and revisions. In [25], two bottom-up and tableau-based approaches are introduced for finding Minimal Unsatisfiability Preserving Sub-terminologies (MUPS). The calculation of MUPS helps in finding and removing the set of axioms in a knowledge base that conclude in inconsistencies. In [20], the repair of inconsistent ontologies is analyzed and solutions for improving the explanation of inconsistencies are explored. More recently, concepts of laconic and precise justifications have been introduced to complement the latter approach [18].

The issue of update becomes even more prominent in modular representations of ontologies, in light of the fact that an update in one ontology module may lead to unintended consequences in other ontology modules due to their coupling. Nonetheless, very few work focus on analyzing the impact of update and revision in modular ontologies [27].

Our contribution in this paper is to analyze the notion of updates in modular ontologies and specially within the context of the IBF representation. We propose a principle for updating modular ontologies, which states that the entailment of an updated ontology about its internal elements must be kept constant when a change takes place in a foreign but related ontology. We explore different scenarios for ABox and TBox update propagation in IBF modular ontologies through different examples and provide a method for computing TBox updates.

The rest of the paper is organized as follow: The next section gives some preliminaries of description logics. We give an introduction to the interface-based formalism for modular ontologies in Section 3. In Section 4, we analyze ABox and
The Effect of Modification and Update in IBF modular ontologies. Section 5 provides a method for updating IBF modular ontologies with TBox modifications. Finally, Section 6 concludes the paper.

2 Preliminaries

A DL knowledge base is defined as $\Psi = \langle T, A, R \rangle$, where $T$ denotes TBox and comprises of a set of general inclusion axioms and $A$ stands for ABox and comprise of a set of instance assertions. A DL-RoleBox, $R$, is comprised of a set of subsumption and inversion relationships between roles. The signature of an ontology is defined as a set of all concept names $(C_N)$, role names $(R_N)$ and individuals $(I_N)$ which are included its knowledge base.

The signature of $\psi$ is a disjoint union $\text{SIG}(\psi) = \text{SIG}_{ext}(\psi) \sqcup \text{SIG}_{int}(\psi)$ of its external and internal elements. The semantic of a DL is defined by an interpretation $I = (\Delta^I, \cdot^I)$ where $\Delta^I$ is a non-empty set of individuals and $\cdot^I$ is a function which maps each $C \in C_N$ to $C^I \subseteq \Delta^I$, each $R \in R_N$ to $R^I \subseteq \Delta^I \times \Delta^I$ and each $a \in I_N$ to an $a^I \in \Delta^I$. An interpretation $I$ is a model of a TBox $T$, if for every inclusion axiom like $C \sqsubseteq D$ in $T$, we have $C^I \subseteq D^I$. A TBox is consistent iff it has a model. A concept $C$ is satisfiable if there is a model $I$ for $T$ such that $C^I \neq \emptyset$.

3 Interface-Based Formalism for Modular Ontologies (IBF)

In this section, we give an introduction to the interface-based formalism for modular ontologies [10, 8, 9, 7]. The two drivers of the formalism are 1) the exploitation of the notion of interfaces in modular ontologies; 2) the utilization of queries$^1$ for creating a mapping between individuals of different modules. An interface is a set of concept and role names and their inclusion axioms. A module is an ontology in any description logic language, which can utilize or realize a set of interfaces. A module utilizes an interface if it uses the concepts and roles of the interface in its local knowledge base, according to the TBox axioms of that interface. A module realizes an interface when it provides definitions and more specialized inclusion axioms as well as appropriate ABox assertions for the interface concepts and roles. Different modules and interfaces can be configured to form a meaningful modular ontology, in the sense that for every module which utilizes an interface, a realizer module is assigned to it. The formalism uses queries to allow a utilizer module employ the local knowledge base of the others. In the following, we precise the definition of the formalism.

**Definition 1** An interface $I$ is defined as $I = (C_N, R_N, T)$ where $T$ is the TBox of the interface and $C_N$ and $R_N$ are sets of concept and role names used in $T$. $I$ has no ABox assertions.

---

$^1$ In our proposal in [10], we made use of epistemic queries [6]
Definition 2: An ontology module $M$ is defined as $M = \langle \Psi, I_r, I_u \rangle$ where $\Psi$ is the knowledge base and $I_r$ is a set of all interfaces which is realized by $M$ and $I_u$ is a set of all interfaces which is utilized by $M$. $M$ can be in any description logic language, but it must support nominals.

Let $P$ be a concept or role name in an interface $I$, it is referred to in the knowledge base of modules as $I : P$. A module $M$ realizes an interface $I$ iff $I \in I_r$.

Given an interface, we refer to the module which uses it as the utilizer module and the module which gives semantics to its terms as the realizer module. As an example, suppose that we want to develop a modular ontology for the tourism domain. We can define the modular ontology as a set of ontology modules: Tourism and North American Destination and the interface: $I$. The ontology module Tourism has information about different tourism destinations, tourist-related activities and also accommodations. The North American Destination ontology module includes and categorizes the name of North American cities, states/provinces and areas. The Tourism ontology module needs to utilize the geographic information of the North American Destination ontology module for providing the address of its destinations. The interface $I$ is used to connect these two ontologies. The Tourism ontology module utilizes the $I$ interface while North American Destination realizes the interface and its concepts by providing their meaning, properties and instances. Figure 1 shows this example.

A module which utilizes an interface needs to access the instances provided by the realizer modules. In the interface-based formalism, we follow a query-based approach to augment the semantic of a utilizer module with the individuals provided by the realizer modules. For instance, regarding our example about the Tourism modular ontology, the Tourism ontology module may pose a query to the North American Destination ontology module about the city of an accommodation. Through the proposed formalism, the Tourism KB is augmented with the individuals that are provided by North American Destinations for the City concept.
Based on the definition of interfaces and modules, we now define a modular ontology as follows:

**Definition 3** A modular ontology is a triple \( O = (M, I, F) \) where \( M \) is a set of ontology modules, \( I \) is a set of interfaces and \( F \) is a configuration function \( F : M \times I \rightarrow M \) which chooses one realizing module for every utilizer module-interface pair. \( F(M, I) = M' \) if:

1. \( I \in I_u^M \) and \( I \in I_u^{M'} \),
2. The union of \( M \) and \( M' \) with the knowledge base of \( I \) be consistent,
3. Let \( C_i \) and \( R_j \) be the result sets of queries \( I : C_i \) and \( I : R_j \) posed to \( M' \), \( \Psi_M \bigcup_{C_i \in I} (I : C_i \equiv C_i) \bigcup_{R_j \in I} (I : R_j \equiv R_j) \) is consistent.

Based on Definition 3, the final form of a modular ontology is specified by the configuration function \( F \). This function shows the connected modules through interfaces and its value can be set at configuration time. For being connected through the configuration function, two ontology modules should satisfy the three conditions mentioned in Definition 3. First of all, a module should realize an interface in order to be selected by the configuration function and be connected to the utilizer module. Second, the union of the knowledge base of both ontology modules with the knowledge base of the interface should be consistent. And finally, the third condition ensures that the integration of two modules does not entail inconsistencies. Since the domain of the utilizer module would be augmented by the individuals of the interface terms from the realizer modules through queries, condition three ensures that this augmentation does not lead to an inconsistency in the utilizer module.

### 4 Update Propagation in IBF

In this section, we analyze the impact of change in an ontology module or an interface on the other connected ontology modules within the context of the IBF ontology modularization. We first show examples of two possible scenarios for change propagation in IBF modular ontologies and then provide formal definitions for the notions of modification and update.

Let's first suppose that the knowledge base of the Location interface, within the Tourism modular ontology discussed in the previous section, contains the following axioms:

1. \( \text{Small-City} \sqsubseteq \text{City} \)
2. \( \text{Expensive-City} \sqsubseteq \text{City} \)
3. \( \text{Domain(hasHotel)=City} \)
4. \( \text{Inv(hasHotel) = hasBranchIn} \)

According to the axioms of the above knowledge base, the Location interface asserts that \text{Small-City} and \text{Expensive-City} are two subclasses of the concept...
City. In addition, the Location interface has two properties: hasHotel and has-
BranchIn, which are defined as each others’ inverses. Furthermore, a City may
be connected to an individual by the hasHotel property.

Assume that the knowledge base of the Tourism ontology module has been
defined as the following. Note that the interface concepts and roles in the Tourism
ontology module are referred to using prefix I.

1. Economic-Hotel ⊑ Hotel
2. Domain(HasPrice)=Hotel, Range(HasPrice)=Price-For-Hotel
3. Less-Than-50-Dollars ⊑ Price-for-Hotel
4. I:Small-City ⊑ ∀I:hasHotel.Economic-Hotel ⊓ Easy-Road-Transporting
5. Economic-Hotel ⊑ ∃hasPrice.Less-Than-50-Dollars
6. ∃hasPrice.Less-Than-50-Dollars ⊑ ∀I:hasBranchIn.¬I:Expensive-City

Accordingly, in the Tourism ontology module Hotel and Price-For-Hotel
are connected through the property hasPrice. In addition, it has been asserted
that all hotels in Small-Cities are Economic-Hotel, where an Economic-Hotel
has a price less than 50 dollars per night. Furthermore, I:Small-City from the
point of view of the tourism ontology module is a sub-concept of the places that
provide easy road transportation (Easy-Road-Transporting). In addition, the
knowledge base asserts that those hotels that have a price less than 50 dollars
can never be found in expensive cities.

The following examples show two cases where updates in the Location inter-
face and North American Destinations ontology module induce inconsistencies
in the Tourism ontology module.

**Example 4 TBox modification**

Assume that the Location interface has been updated such that it asserts that
a Small-City has at least a hotel that has a branch in an expensive city as well:

Small-City ⊑ ∃hasHotel.∃hasBranchIn.Expensive-City

Now, if we make the assumption that Fredericton is a small city, Toronto
is an expensive city, and Delta Hotel is a hotel, Fredericton has a Delta Ho-
tel, which also has a branch in Toronto. This update retains the consistency
of the Location interface; However, it conflicts with the knowledge base of the
Tourism ontology and makes the I:Small-City concept in the tourism ontology
unsatisfiable and hence, causes the ontology to become inconsistent.

The next example depicts the type of inconsistencies that arise when the
ABox of the realizer module has undergone change.

**Example 5 Abox modification**

Assume that the North American Destinations ABox has been updated such
that it includes the following assertions:

Small-City(Honolulu)
Expensive-City(Honolulu)
After the re-augmentation of the Tourism ontology module with its realizer (i.e., the North American Destinations ontology) it becomes an inconsistent knowledge base. The reason is that the Tourism ontology entails a new subsumption between two interface concepts $I$:Small-City $\sqsubseteq \neg I$:Expensive-City. The new update describes a city, Honolulu, which is both small and expensive which is conceptually inconsistent with the previous axioms.

Now, definitions 6 and 7 provide formal platforms for the notion of modification in the TBox and ABox of IBF modular ontologies.

**Definition 6** Let $I$ be an interface and $M_u$ and $M_r$ be two ontology modules in a modular ontology such that $F(M_u, I) = M_r$. $\mu_{T,I}$ is a TBox modification for $M_u$ iff it is comprised of a set of inclusion axioms such as $\alpha \sqsubseteq \beta$, where $\text{SIG}(\alpha) \subseteq \text{SIG}(I)$, $\text{SIG}(\beta) \subseteq \text{SIG}(I)$ and also $M_r \cup \mu_{T,I}$ and $I \cup \mu_{T,I}$ are consistent.

**Definition 7** Let $I$ be an interface and $M_u$ and $M_r$ be two ontology modules in a modular ontology such that $F(M_u, I) = M_r$. $\mu_{A,I}$ is an ABox modification for $M_u$ iff it is comprised of a set of ABox assertions in the form of $C(a)$ or $R(a,b)$, where $C \in C_N(I)$, $R \in R_N(I)$, $a$ and $b$ be individuals in $M_r$ and also $M_r \cup \mu_{A,I}$ is consistent.

Examples 4 and 5 show that a modification may induce inconsistencies in an ontology module. For updating an ontology module with a modification, a procedure must resolve any consequential inconsistencies. Through Definition 8, we define an update operator for TBox and ABox modifications. For defining this operator we apply the principle of zero change for internal elements. This principle means that we attempt to apply the incoming change and resolve the consequential inconsistencies in such a way that the knowledge about the internal concepts and roles of an ontology module do not experience any update. Instead, those axioms that refer to interface concepts or roles need to be changed in order to achieve consistency. The hypothesis behind this principle is that the new information about the interface elements must only effect the interface elements of the knowledge base.

As an explanatory example, consider the case of Example 5, if the fifth axiom, Economic-Hotel $\sqsubseteq \exists I: \text{hasPrice. Less-Than-50-Dollars}$, is removed from the knowledge base, the inconsistencies within the ABox assertions about Honolulu will be resolved; however, according to our principle, the axiom that an Economic-Hotel has a price less that 50 dollars is an internal knowledge of the Tourism ontology and is not dependent on the Location interface which is realized by the North American Destination ontology; therefore, although the removal of Economic-Hotel $\sqsubseteq \exists I: \text{hasPrice. Less-Than-50-Dollars}$ entails consistency, the principle of zero change for internal elements prevents it from being removed. This principle protects the internal information of an ontology module against inconsistency resolution procedures performed as result of inconsistency caused by the other parts of the modular ontology.
Definition 8 Let $M_u$ be an ontology module in a modular ontology which utilizes interface $I$, $\psi(M_u)$ be the knowledge base of $M_u$, $\psi(I)$ be the knowledge base of the interface and $\mu = \mu_{T,I}[\mu_{A,I}]$ be an ABox[TBox] modification for $M_u$. Furthermore, let $Cn(KB)$ be the set of all logical entailments of the knowledge base $KB$, $\psi'$ is the result of an update on $\psi(M_u)$ using $\mu$ and is denoted by $\psi' = \psi(M_u) \circ \mu$ iff:

i) $\psi' \cup \psi(I)$ is consistent,

ii) $Cn(\mu) \subseteq Cn(\psi' \cup \psi(I))$

iii) $\forall \alpha, s.t. \text{SIG}(\alpha) \subseteq \text{SIG}_{\text{IN}}(\psi), \alpha \subseteq Cn(\psi' \cup \psi(I))$, 

iv) Let $\Delta = \{\alpha | \text{SIG}(\alpha) \cap \text{SIG}(I) \neq \emptyset, \alpha \in Cn(\psi' \cup \psi(I)) \text{ and } \alpha \notin Cn(\psi' \cup \psi(I))\}$ then $|\Delta|$ must be $\leq$ minimum.

In the above definition, (i) ensures that the resulting knowledge base after update is consistent. Since the knowledge base of the utilizer module must be augmented by the knowledge base of its interface (see [10]) after any update, we consider the union of $\psi'$ and the knowledge base of the interface in all of following items. Hence, for example in the case of (i), not only $\psi'$, but also its union with $\psi(I)$ must be consistent in order to prevent any inconsistency when augmenting the utilizer module. (ii) guaranties that the union of $\psi'$ and the knowledge base of the interface that it utilizes entails all of axioms in the modification. Observably, the update operator does not remove any piece of information from $\mu$. (iii) enforces the principle of zero change for internal elements. Finally, (iv) ensures that the principle of minimal change is observed for interface elements.

In the next section, we provide a method for updating an ontology module with TBox modifications. In [11], we have already provided a framework for updating modular ontologies with ABox modifications.

5 TBox Update Procedure

In order to update an ontology module with TBox modifications, we exploit the notion of diagnosis in knowledge bases for finding those problematic axioms in an ontology module which give rise to inconsistencies when integrated with a TBox modification. Clearly, the removal of these problematic axioms will make the ontology module consistent. Definition 9 provides formal definition for diagnosis in IBF modular ontologies. According to this definition, a diagnosis for an ontology module is a subset of its knowledge base whose removal makes the integration of the ontology module, the incoming modification and the knowledge base of the utilized interfaces consistent.

Definition 9 Let $M_u$ be an ontology module in a modular ontology that utilizes the interface $I$, $\psi(I)$ be the knowledge base of $I$, $\psi(M_u)$ be the knowledge base of $M_u$ and $\mu_{T,I}$ be a TBox modification for $M_u$. $T' \subseteq \psi(M_u)$ is a diagnosis for $M_u$ iff $\psi(M_u) \cup \mu_{T,I} \cup \psi(I) - T'$ is consistent. The set of all diagnoses for the ontology module $M_u$ and the modification $\mu_{T,I}$ is denoted as $\text{Digns}(M_u, \mu_{T,I})$. 

With regards to the principle of zero change for internal elements, we are interested in those diagnoses that include interface elements. Consequently, we define external diagnoses as those which have at least one interface element in each of their axioms.

**Definition 10** Let \( Digns(M_u, \mu T, I) \) be the set of all diagnoses of the ontology module \( M_u \) and the modification \( \mu_T, I \). \( D \in Digns(M_u, \mu_T, I) \) is an external diagnosis for \( M_u \) iff for all of the axioms such as \( \alpha \in D \), \( \text{SIG}(\alpha) \cap \text{SIG}(I) \neq \emptyset \). The set of all external diagnoses for the ontology module \( M_u \) is denoted as \( Digns_{ext}(M_u, \mu_T, I) \).

The following theorem shows that there is at least one external diagnosis for an ontology module in an inconsistent modular ontology when it is integrated with a TBox modification.

**Theorem 11** Let \( M_u \) be an ontology module in an IBF modular ontology that utilizes the interface \( I \), \( \psi(I) \) be the knowledge base of \( I \), \( \psi(M_u) \) be the knowledge base of \( M_u \) and \( \mu_T, I \) be a TBox modification for \( M_u \). There exists at least one external diagnosis for \( M_u \) if \( \psi(I) \cup \psi(M_u) \cup \mu_T, I \) is inconsistent.

**Proof.** Let \( T' \subseteq \psi \) be the set of all axioms in \( \psi \) that have at least one interface concept or role. We show that \( \psi - T' \cup \mu_T, I \cup \psi(I) \) is consistent. From Definition 6, we know that \( \mu_T, I \cup \psi(I) \) is consistent. From the definition of the TBox modification, we can also infer that \( \text{SIG}(\mu_T, I) \subseteq \text{SIG}(I) \). Since \( \psi - T' \) does not have any reference to interface concepts and roles and \( \text{SIG}(\mu_T, I \cup \psi(I)) \cap \psi - T' = \emptyset \). Consequently, \( \psi - T' \cup \mu_T, I \cup \psi(I) \) is consistent.

For resolving inconsistencies, an efficient method must find and remove the minimal set of external diagnosis due to the principle of minimal change. Definition 12 characterizes minimal external diagnoses.

**Definition 12** Let \( Digns_{ext}(M_u, \mu_T, I) \) be the set of all external diagnoses of the ontology module \( M_u \) and the modification \( \mu_T, I \). An external diagnosis \( D \in Digns_{ext}(M_u, \mu_T, I) \) is minimal iff there is no proper subset \( D' \subseteq D \) such that \( D' \) is an external diagnosis.

There are different methods in the literature that find proper diagnosis for description logic databases [25, 14]. [19] provides a method for obtaining diagnoses of a \( SHI^{ON} \) knowledge base. [26] introduces a divide and conquer method that obtains the set of minimal diagnosis for a knowledge base \( KB \) and a constant part \( B \), using the existing methods for determining minimal diagnosis for a monolithic knowledge base. In this paper, we do not go through the details of finding a diagnosis. Assuming that the set of diagnosis is given, we focus on finding external diagnosis and a method for updating ontology modules such that they satisfy the conditions introduced in Definition 8.

**Example 13** With regards to Example 4, the following is the list of all minimal diagnoses:
The set of minimal external diagnoses is equal to \{\{\text{Axiom } \#4\}, \{\text{Axiom } \#6\}\}

The next step in the inconsistency resolution procedure is to remove the minimal external diagnosis from the knowledge base and produce a consistent integrated knowledge base. For example, in the case of Example 13, removing Axiom \#4 makes the ontology module consistent. However, in the current representation of the knowledge base, the removal of Axiom \#4, removes some other non-problematic pieces of information. By removing Axiom \#4 (I:Small-City ⊑ ∀I:hasHotel.Economic-Hotel ⊓ Easy-Road-Transporting), the fact that I:Small-City is a sub-concept of Easy-Road-Transporting will be removed, even though this information does not cause any inconsistency and does not have any conflict with the TBox modification. To address this kind of issue and in order to produce axioms that are as small and flat as possible, we use \(\delta(\psi)\), where \(\delta\) is a satisfiability preserving structural transformation on \(\psi\) [18]. This structural transformation process removes all nested descriptions to reach its desired goal. We employ the following set of transformation rules from [18] in order to produce a flat equivalent of the original knowledge base:

\[
\delta(\psi) := \bigcup_{a \in R \cup A} \delta(\alpha) \cup \bigcup_{C_1 \subseteq C_2 \in T} \delta(\top \subseteq \text{nnf}(\neg C_1 \cup C_2))
\]

\[
\delta(D(a)) := \delta(\top \subseteq \neg\{a\} \cup \text{nnf}(D))
\]

\[
\delta(\top \subseteq C \cup D) := \delta(\top \subseteq A_D \cup C) \cup \bigcup_{i=1}^{n} \delta(A_D' \subseteq D_i) \text{ for } D = \bigcap_{i=1}^{n} D_i
\]

\[
\delta(\top \subseteq C \cup \exists R.D) := \delta(\top \subseteq A_D \cup C) \cup \{A_D \subseteq \exists R.A_D'\} \cup \delta(A_D' \subseteq D)
\]

\[
\delta(\top \subseteq C \cup \forall R.D) := \delta(\top \subseteq A_D \cup C) \cup \{A_D \subseteq \forall R.A_D'\} \cup \delta(A_D' \subseteq D)
\]

\[
\delta(\top \subseteq C \cup \geq nR.D) := \delta(\top \subseteq A_D \cup C) \cup \{A_D \subseteq \geq nR.A_D'\} \cup \delta(A_D' \subseteq D)
\]

\[
\delta(\top \subseteq C \cup \leq nR.D) := \delta(\top \subseteq A_D \cup C) \cup \{A_D \subseteq \leq nR.A_D'\} \cup \delta(A_D' \subseteq D)
\]

\[
\delta(A_D' \subseteq D) := \delta(\top \subseteq \neg A_D \cup D) \text{ (If } D \text{ is not of the form } A \text{ or } \neg A)
\]

\[
\delta(\beta) := \beta \text{ for any other axiom}
\]

where \(\text{SIG}(A) \subseteq \text{SIG}(\psi)\), \(A_D\) and \(A_D'\) are fresh concepts names: \(\text{SIG}(A_D) \cap \text{SIG}(\psi) = \emptyset\) and \(\text{SIG}(A_D') \cap \text{SIG}(\psi) = \emptyset\). \(C_i\) and \(D\) are arbitrary concepts, excluding \(\top\) and \(\bot\) and any fresh literal concept. \(C\) is a disjunction of arbitrary concepts that can be empty. The transformation ensures that concept names in \(\text{SIG}(M_a)\) only appear in the form of \(X \subseteq A\) or \(X \subseteq \neg A\), where \(A\) is an atomic concept and \(X\) is not in \(\text{SIG}(M_a)\).

For updating the ontology module, we assume that the knowledge base of the ontology module has been transformed using the above rules and its minimal external diagnosis has been specified. Because of this transformation, issues such as the ones mentioned earlier (the removal of Axiom \#4 in Example 13) will be prevented. This is due the fact that all axioms are in the form of \(X \subseteq A\) or \(X \subseteq \neg A\).
The next step is to remove the set of diagnoses in such a way that the principle of zero change for internal elements and the principle of minimal change are satisfied. Algorithm 1 illustrates our proposed procedure for updating ontology modules in IBF modular ontologies. This algorithm has two important segments:

1) Lines 11 to 13 that find all internal entailments of the ontology module before updating and inserting them into the new updated knowledge base. This segment guarantees the principle of zero change for internal elements. Even though we only remove external diagnoses, the resulting knowledge base may miss some of its entailments with regards to its internal concepts and roles. Example 14 shows a simple case of this situation.

**Algorithm 1**: The algorithm for updating ontology modules with TBox modifications.

**Input:**
\[ \psi(M_u) : \text{the knowledge base of the ontology module } M_u \text{ in an IBF modular ontology} \]
\[ \psi(I) : \text{the knowledge base of the interface which is utilized by } M_u \]
\[ \mu_{T,I} : \text{the TBox modification} \]
\[ Digns_{ext}(M_u, \mu_{T,I}) : \text{the set of all minimal external diagnoses of the ontology module } M_u \text{ for the modification } \mu_{T,I} \]

**Output:** \( \Psi' \): a set of updated knowledge bases

1. if \( \psi(M_u) \cup \psi(I) \cup \mu \) is consistent then
   2. \( \Psi' = \{ \psi(M_u) \} \)
   3. return
   4. end

5. else
   6. foreach \( D \in Digns_{ext}(M_u, \mu_{T,I}) \) do
     7. \( S := \psi(M_u) - D \cup \psi(I) \cup \mu_{T,I} \)
     8. \( \Delta := \text{Cn}(\psi(M_u) \cup \psi(I)) \setminus \text{Cn}(S) \)
     9. \( K = \emptyset \)
     10. foreach \( \alpha \in \Delta \) do
         11. if \( \text{SIG}(\alpha) \subseteq \text{SIG}_{int}(M_u) \) then
             12. \( S := S \cup \alpha \)
             13. end
         14. else
             15. \( K := K \cup \alpha \)
             16. end
         17. end
     18. \( K' := \text{The largest subset of } K \text{, s.t., } K' \cup S \text{ is consistent.} \)
     19. \( S := S \cup K' \)
     20. \( \Psi' := \Psi' \cup S \)
     21. end
     22. end
23. end
Example 14 Suppose $M_u$ be an ontology module which utilizes the interface $I$ and its knowledge base is as follows:

1. $A \sqsubseteq I:D$
2. $I:D \sqsubseteq B$
3. $B \sqsubseteq \neg I:C$

Assume that a TBox modification $\mu_{T,1}$ is equivalent to $\{I:D \sqsubseteq I:C\}$. Axiom #2 is a diagnosis for the knowledge base. However, if we remove Axiom #2, the internal entailment $A \sqsubseteq B$ will also be removed, which is undesirable.

Because of the existence of these internal entailments, in our algorithm, we find all internal entailments and insert them into the updated knowledge base.

2) Lines 14 to 19 that find the set of entailments of the ontology module about its interface concepts and roles and inserts them into the updated knowledge base if their insertion does not lead to inconsistency. This part assures the principle of minimal change. Since we are using a transformed version of the knowledge base, those entailments that have been missed in the updated version are not numerous. Consequently, checking and inserting them to the updated knowledge base will not lead to performance issues.

6 Concluding Remarks

In this paper, we have analyzed the notion of update in modular ontologies. Through different examples, we illustrated that a modification on a knowledge base may lead to inconsistencies in different connected knowledge bases. This may happen even in cases where the modification can be cleanly integrated in the original knowledge base without any inconsistencies, but cause ripple effects on other connected ontologies. The main contributions of this paper are as follow:

- The development of a basis for modification and update in modular ontologies. This foundation provides the necessary definitions for diagnosis in modular ontologies considering the internal and external ontological elements.
- The introduction of the principle of zero change for internal elements for updating ontologies in modular representations. According to this principle, any modification in a foreign knowledge base, which has propagated to the current ontology module, must not change the internal elements of this ontology module.
- The proposal of an algorithm for updating IBF modular ontologies with TBox modifications. This algorithm respects the principles of zero change for internal elements and minimality of change.

As future work, we intend to generalize the results of our current research for handling ABox and TBox modifications for supporting other types of ontology modularization formalisms. We believe that the proposed principle and the supporting definitions are a strong basis for dealing with inconsistency in updating modular ontologies and hence can be beneficial for other formalisms as well.
References


8. Ensan, F. Formalizing ontology modularization through the notion of interfaces. In 16th International Conference on Knowledge Engineering and Knowledge Management Knowledge Patterns (EKAW'08) (2008), pp. 74–82.


13. Flouris, G., Plexousakis, D., and Antoniou, G. On Applying the AGM Theory to DLs and OWL. LECTURE NOTES IN COMPUTER SCIENCE 3729 (2005), 216.


