Automatic Building of an Appropriate Global Ontology

Cheikh Niang[1,2], Béatrice Bouchou[1], Moussa Lo[2], and Yacine Sam[1]

[1] Université François Rabelais Tours - Laboratoire d'Informatique
{first.last}@univ-tours.fr
[2] Université Gaston Berger de Saint-Louis - LANI
{first.last}@ugb.sn

Abstract. Our objective is to automatically build a global ontology from several data sources, annotated with local ontologies and aiming to share their data in a specific application domain. The originality of our proposal lies in the use of a background knowledge, i.e. a reference ontology, as a mediation support for data integration. We represent ontologies using Description Logics and we combine syntactic-matching with logical-reasoning in order to build the shared global TBox from both the TBoxes of sources and that of the reference ontology.

Keywords: Background Knowledge, Data Integration, Descripion Logic.

1 Introduction

As the need for Web Data Integration is still growing, we address here the first challenge pointed out in [15]: "How to build an appropriate global schema". Indeed, many organizations hold some similar data in specific domain and want to share some parts of it. Data integration may alleviate users from knowing the structure of different sources, as well as the way they are conciliated, when making queries [15].

When the access to heterogeneous data sources is made possible using ontologies, the integration process in called *ontology-based data integration*. Ontologies offer a formal semantics which allows the automation of tasks such as heterogeneity resolution, consistency checking, inferrence, etc. There are three main ontology-based data integration architectures in the literature [21], namely *(i)* the *single-ontology*, *(ii)* the *multiple-ontologies* and *(iii)* the *hybrid* approaches. In the first one, all the data sources are related to a global ontology: this approach requires from all the sources the same view of the domain, for instance the same granularity-level, because in the presence of sources with a different view of the domain, finding a consensus in a minimal ontology commitment is a difficult task. This approach is implemented for instance in [3].

In the *multiple-ontologies* approach, for instance in the OBSERVER system [16], each data source is described with its own (local) ontology, and inter-ontology mappings must be defined for interoperability. The lack of a common vocabulary between the sources makes this task difficult. The *hybrid* approach combines the two precedent ones, allowing to overcome their drawbacks by defining a global shared vocabulary in addition to local ontologies: [11] is an example of such an integration architecture.

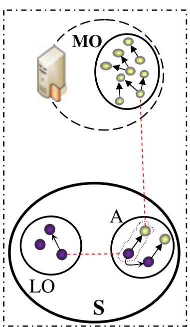
In this paper we build on the hybrid approach as we propose to automatically build a global ontology from local ones. As usual in data integration systems, our global

J. Eder, M. Bielikova, and A.M. Tjoa (Eds.): ADBIS 2011, LNCS 6909, pp. 429–443, 2011.

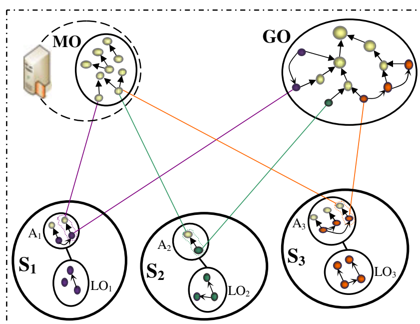
ontology must be linked to local ontologies by mappings. The two basic solutions for doing so are the LAV (local as view) and the GAV (global as view) mappings. Each of them has advantages and drawbacks: LAV approach allows to define the global ontology independently from the sources, so adding or removing a new data source is easy but query processing is harder. Query processing is less complex with GAV approach, since the global ontology is defined from the data sources, but sources must be known in advance and adding new data source is not easily supported.

We propose to overcome some drawbacks of both hybrid approach and GAV mappings *by using a background-knowledge*, represented by a reference ontology, in order to automatically build a global ontology from local sources. We call reference ontology an ontology developed independently from any specific objective by experts in knowledge engineering with the collaboration of domain experts. It is a robust conceptualization of the knowledge about a given generic domain such as medicine, tourism, agriculture, etc. AGROVOC[1] and NALT[2] in the agriculture domain and MeSH[3] in the medical field are some examples of reference ontologies. The growth of Semantic Web allows to expect that such reference ontologies will become more and more accessible and usable by machines in the coming years.

The algorithm presented in this paper follows the mediation-based process illustrated in Fig. 1: each source (S_i) involved in the sharing process is represented by its local ontology (LO_i) and the reference ontology (MO) allows to find the portion of knowledge that each source can share with others. This portion is called agreement (A in Fig. 1(a)). Then each agreement is incrementally integrated in the global ontology (GO) via MO in what we call the conciliation phase (Fig. 1(b)).



(a) Agreement



(b) Conciliation

Fig. 1. General overview of our mediation-based process

The challenge that we point out is: how to automatically build an appropriate global ontology for several data source owners that want to share parts of their data for a specific web application, but that do not want to (or can not) invest much efforts on the hard task of building a consensual appropriate shared conceptual level ? An appropriate

[1] http://www.fao.org/agrovoc

[2] http://agclass.nal.usda.gov/agt

[3] http://www.nlm.nih.gov/mesh/

global ontology in such a sharing context should provide an appropriate conceptualization of the application domain (maximizing relevant information for the sharing process and minimizing irrelevant one). It must allow to add easyly new data sources, and also to remove or update sources. It must allow an easy querying of sources. Finally it must be automatically built and maintained.

This is what our algorithm builds. **For an easy query processing** it lies on the GAV approach. However it generalizes existing proposals so that it is no longer necessary to have sources known in advance. An anchoring phase allows each source to participate in the global ontology to some extent, whatever it is. **For scalability**, it incrementally integrates data sources, so it is easy to add a new source involved in the sharing process. **For an appropriate conceptualization**, it selects in the reference ontology the smallest relevant information portion and, in each data source involved in the sharing process, it selects only information that is relevant to be shared in the application domain. **For automation**, we use Description Logics to represent ontologies. Description Logics (or DL) are formalisms for conceptual representations which have already been successfully used for *(i)* linking Data to Ontologies [17] and *(ii)* building Data Integration Systems [7]. Our choice of DL is mainly motivated by their capability to represent hierarchies and to automaticaly reason on these relationships. Moreover, the inference capabilities of DL are not limited to hierarchies (they are equipped with a formal *logic*-based semantics), so DL are fully justified here as a data model that allows inferences.

The rest of this article is organized as follows: in Section 2 we addres related works, in Section 3 we define the notions used in our algorithm, in Sections 4 and 5 we present our global-ontology-building process, and Section 6 concludes.

2 Related Works

Our approach deals with ontology-based data integration. Very close to our interests are the works based on (i) Description Logics and (ii) Ontology Matching.

In [8] and [19], a formalism for reasoning with multiple local ontologies connected by directional semantic mappings is presented, in other words they introduce the notion of distributed description logics, useful for linking different data sources. The approach presented in [13] is another solution exploiting description logics, namely the $\mathcal{E}$-connections framework, to link different sources for an integration purpose. Both of these approaches have successfully shown the interest of using description logics for efficiently exploiting distributed data sources. The contribution of our proposal is the introduction of automation in the linking process. We use for this a reference ontology: on the one hand links between source ontologies are obtained from the taxonomical relationships of the reference ontology. On the other hand, mappings between the global ontology and sources are obtained by syntactic-matching, from source-concepts' names to reference-ontology-concepts' names.

For that reason, our algorithm depends on the performance of Ontology Matching techniques (cf. Section 4), which constitute a very active research field (see [20] for a survey). The use of reference ontologies has been investigated in this field, see for example [2], [1] and [18]. It was shown that the reference ontology can significantly improve the performance of the matching process. The contribution of our proposal is to show that the reference ontology also allows to enrich the semantics of links discovered in the matching process. As an example, we can have in a local ontology two anchored concepts, e.g. *Onion* and *Tomato*, not related: our conciliation algorithm can relate them via a common ancestor of their anchor concepts, e.g. *Vegetable* (see Fig. 7).

Finally, our work is related to data integration systems using GAV mapping ([15], [6] and [12]) and our contribution here is again the use of the reference ontology. It stands for the information about sources that allows adding easily new sources.

3 Preliminaries

In our approach, ontologies are expressed in Description Logics (DLs) [4], a family of logic-based representation formalisms. They allow representing the domain of interest in terms of *concepts*, denoting sets of objects, and *roles*, denoting binary relations between (instances of) concepts [17]. A DL ontology consists of a *TBox* (Terminological Box) and an *ABox* (Assertional Box): the former formally specifies concepts and roles and the latter represents their instances. DLs differ in constructs they allow to specify concepts and roles. In this paper, we consider the *DL-Lite*$_{\mathcal{A}}$ description logic [17]. It is known as one of the most expressive DL in the *DL-Lite* family [9].

3.1 *DL-Lite*$_{\mathcal{A}}$ Syntax and Semantics

The syntax of *DL-Lite*$_{\mathcal{A}}$ expressions is defined as follows [17]:

$$
\begin{array}{ll}
B ::= A \mid \exists Q \mid \delta(U_C) & Q ::= P \mid P^- \\
C ::= \top_C \mid B \mid \neg B \mid \exists Q.C & R ::= Q \mid \neg Q \\
E ::= \rho(U_C) & \\
F ::= \top_D \mid T_1 \mid \ldots \mid T_n & V_C ::= U_C \mid \neg U_C
\end{array}
$$

Where A denotes an *atomic concept*, i.e., a concept denoted by a name, B a *basic concept*, C a *general concept*, and $\top_C$ the *universal concept*. E denotes a basic value-domain, i.e., the range of an attribute, F a *value-domain expression*, and $\top_D$ the *universal value-domain*. P denotes an *atomic role*, Q a *basic role*, and R a *general role*. U_C denotes an *atomic attribute* and V_C *a general attribute*.

The semantics of every *DL* expression is specified in term of its first-order interpretation. An *interpretation* is defined as a pair $I = (\Delta^I, \cdot^I)$, where Δ^I is the domain interpretation and $\cdot^I$ an *interpretation function*. In *DL-Lite*$_{\mathcal{A}}$, Δ^I is composed of two non-empty sets: Δ^I_O, the *domain of objects*, i.e. the set of all allowed objects in the domain, and Δ^I_V, the *domain of values*, i.e. the set of all allowed values in the domain ($\Delta^I = \Delta^I_O \cup \Delta^I_V$). The interpretation function assigns a subset of Δ^I to each concept or value domain, and a subset of $\Delta^I \times \Delta^I$ to each role or attribute, in such a way that the following conditions are satisfied:

$$
\begin{array}{ll}
\top^I_C = \Delta^I_O & P^I \subseteq \Delta^I_O \times \Delta^I_O \\
\top^I_D = \Delta^I_V & U^I_C \subseteq \Delta^I_O \times \Delta^I_V \\
A^I \subseteq \Delta^I_O & (\neg U_C)^I = (\Delta^I_O \times \Delta^I_V) \backslash U^I_C
\end{array}
$$

$(\rho(U_C))^I = \{v \mid \exists o.(o,v) \in U_C^I\}$ $\quad$ $(\exists Q)^I = \{o \mid \exists o'.(o,o') \in Q^I\}$
$(\delta(U_C))^I = \{o \mid \exists o.(o,v) \in U_C^I\}$ $\quad$ $(\neg Q)^I = (\Delta_O^I \times \Delta_O^I)\backslash Q^I$
$(P^-)^I = \{(o,o') \mid (o',o) \in P^I\}$ $\quad$ $(\neg B)^I = \Delta_O^I \backslash B^I$

3.2 Our *DL-Lite*$_{\mathcal{A}}$ Ontologies

A *DL-Lite*$_{\mathcal{A}}$ *ontology* $O = \langle \mathcal{T}, \mathcal{A} \rangle$ specifies a given application domain in terms of a TBox $\mathcal{T}$ representing its intensional part and an ABox $\mathcal{A}$ representing the extensional one. $\mathcal{T}$ consists in a set of *intensional expressions* specified according to the following syntax: $B \sqsubseteq C \mid E \sqsubseteq F \mid Q \sqsubseteq R \mid U_C \sqsubseteq V_C \mid (funct\ Q) \mid (funct\ U_C)$

A concept (respectively, value-domain, role, and attribute) inclusion expresses that a basic concept B (respectively, basic value-domain E, basic role Q, and atomic attribute U_C) is subsumed by a general concept C (respectively, value-domain F, role R, attribute V_C). A role (attribute) functionality expresses the functionality of a role. The semantics of a *DL-Lite*$_{\mathcal{A}}$ TBox is defined by its interpretations. A given interpretation I satisfies:

– a concept (respectively, value-domain, role, attribute) inclusion assertion $B \sqsubseteq C$ (respectively, $E \sqsubseteq F$, $Q \sqsubseteq R$, $U_C \sqsubseteq V_C$), if $B^I \subseteq C^I$ (respectively, $E^I \subseteq F^I, Q^I \subseteq R^I, U_C^I \subseteq V_C^I$)
– a role functionality assertion $(funct\ Q)$, if for each $o_1, o_2, o_3 \in \Delta_O^I$ $(o_1, o_2) \in Q^I$ and $(o_1, o_3) \in Q^I$ implies $o_2 = o_3$
– an attribute functionality assertion $(funct\ U_C)$, if for each $o \in \Delta_O^I$ and $v_1, v_2, \in \Delta_V^I$ $(o, v_1) \in U_C^I$ and $(o, v_1) \in U_C^I$ implies $v_1 = v_2$

I is a *model* of $\mathcal{T}$ if and only if I satisfies all intensional expressions in $\mathcal{T}$. $\mathcal{T}$ is *satisfiable* (or *consistent*) if it has at least one model. In this article, we reason essentially on the structural part of an ontology, i.e., a TBox-level reasoning. Moreover, when considering the mediator ontology we restrict ourselves to specify *atomic concept inclusion* (ACI) expressions. An ACI expression is defined as an inclusion of the form $A \sqsubseteq D$, where A and D are atomic concepts. A finite set of ACI expressions is called an *atomic TBox*.

Definition 3.1. *An ACI is an expression of the form $A \sqsubseteq D$, where A and D are atomic concepts. A finite set of ACIs is called an* ***atomic TBox***.

Finally, $\mathcal{A}$ consists in a finite set of *membership assertions* of the form: $A(a)$, $P(a,b)$ and $U_C(a,b)$. As said before, we don't address this part in the present article.

3.3 Inference Capabilities

One of the traditional inference services provided by DLs is computing subsumption relationships between concepts.

Definition 3.2. *Let $\mathcal{T}$ be a TBox, C and D two concept descriptions. The concept C is subsumed by D w.r.t the TBox $\mathcal{T}$ ($C \sqsubseteq_{\mathcal{T}} D$) iff $C^I \subseteq D^I$ for all models I of $\mathcal{T}$.*

In the present article, we explore subsumption reasoning in order to compute the *deductive closure* of an atomic TBox, defined as follows.

Definition 3.3. *Let $\mathcal{T}$ be an atomic TBox. The deductive closure of $\mathcal{T}$, denoted by $cl\mathcal{T}$, is the TBox inductively defined as follows:*

1. *If $A_1 \sqsubseteq_{\mathcal{T}} A_2$, then $A_1 \sqsubseteq_{cl(\mathcal{T})} A_2$.*
2. *If $A_1 \sqsubseteq_{cl(\mathcal{T})} A_2$ and $A_2 \sqsubseteq_{cl(\mathcal{T})} A_3$, then $A_1 \sqsubseteq_{cl(\mathcal{T})} A_3$.*

3.4 Building an Appropriate *DL-Lite*$_{\mathcal{A}}$ Global TBox

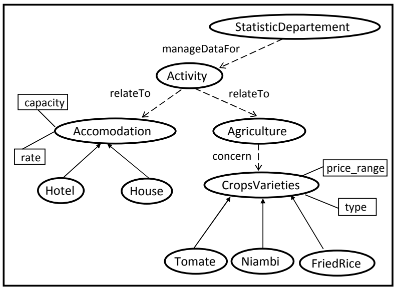
Our objective consists in building a *global ontology*, more precisely a *DL-Lite*$_{\mathcal{A}}$ TBox $\mathcal{T}_g$, based on the local-sources' TBoxes and that of a mediated (reference) ontology. Precisely, the following are the four kinds of TBoxes that we deal with here.

- **The set of local TBoxes** $\{\mathcal{T}_{li}\}$ involved in the sharing process. Each data source S_i is represented by its local TBox $\mathcal{T}_{li}$, denoted LO_i in Fig. 1.
- **The mediator TBox** $\mathcal{T}_m$. It is a *DL-Lite*$_{\mathcal{A}}$ atomic TBox providing general intensional knowledge on the application domain. We consider it as a set of atomic concepts inclusions (*ACI*), *i.e.* a subsumption hierarchy. It is denoted *MO* in Fig. 1.
- **The set of agreement TBoxes** $\{\mathcal{T}_{ai}\}$, denoted A_i in Fig. 1. An agreement TBox $\mathcal{T}_{ai} = \langle \mathcal{T}'_{li}, \mathcal{M}_i \rangle$ is built for each local TBox $\mathcal{T}_{li}$. It is composed of $\mathcal{T}'_{li}$, the subset of $\mathcal{T}_{li}$ containing expressions of $\mathcal{T}_{li}$ that are relevant for the application domain, and $\mathcal{M}_i$, the set of mappings between $\mathcal{T}_{li}$ and $\mathcal{T}_m$.
- **The global TBox** $\mathcal{T}_g = \langle \{\mathcal{T}_{ai}\}, \mathcal{T}'_m \rangle$, denoted *GO* in Fig. 1. It consists in the set of agreement TBoxes $\{\mathcal{T}_{ai}\}$ together with $\mathcal{T}'_m$, which is the smallest subset of $\mathcal{T}_m$ that conciliates every $\mathcal{T}_{ai}$.

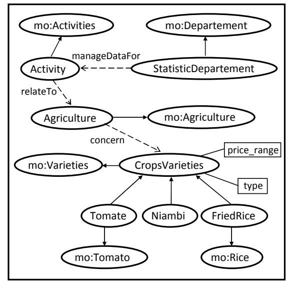
We show how to build $\mathcal{T}_g$ from $\{\mathcal{T}_{li}\}$ by using $\mathcal{T}_m$. It consists in the selection of parts of $\{\mathcal{T}_{li}\}$ to be included in $\mathcal{T}_g$ (Section 4) and then their conciliation (Section 5) in $\mathcal{T}_g$.

4 Agreement

Agreement process consists in the selection of the expressions in $\mathcal{T}_l$ to be included in the global TBox $\mathcal{T}_g$. To identify such knowledge we proceed first by applying an *anchoring* process [2] to select from the local TBox relevant concepts for the application domain. Anchoring consists in associating atomic concepts of a local Tbox, called *anchored concepts*, with concepts of the mediator TBox, called *anchor concepts*. Consider the example shown in Fig. 2, where concepts are represented by ovals, attributes by rectangles and roles by dashed arrows. Single-full arrows represent subsumption relationships between two concepts. Fig. 2(a) shows an excerpt of a local TBox $\mathcal{T}_l$ that deals with both agricultural and accommodation knowledge. We assume that the application domain in which the source $\mathcal{T}_l$ shares its data is the agricultural domain: Fig. 2(b) shows an excerpt of the agreement TBox obtained after the anchoring process. Prefix *mo* : denotes anchor concepts from the mediator TBox $\mathcal{T}_m$. We can notice that only concepts related to agriculture are anchored because no anchor is found for accommodation knowledge. Anchor concepts generalize anchored concepts and will be used for finding semantic links between concepts in different local TBoxes.



(a) Local TBox



(b) Agreement TBox

Fig. 2. Example of the agreement process

We perform two successive anchoring steps: a lexical anchoring process that selects relevant concepts to be anchored, based on syntactic-matching, followed by a semantic one (logical-inference) that selects other ones not detected in the first step.

Lexical Anchoring Process. It consists in matching a local TBox $\mathcal{T}_l$ with the mediator TBox $\mathcal{T}_m$, *i.e.* in computing a set of mappings as defined in [20].

Definition 4.1. *Let $\mathcal{T}_l$ be a local TBox and $\mathcal{T}_m$ be the mediator TBox. Lexical anchoring of $\mathcal{T}_l$ w.r.t $\mathcal{T}_m$ consists in finding a set of mappings $\mathcal{M} = \langle m_1, \ldots, m_n \rangle$ such that each m_i is an assertion of the form: $m_i = A_l \sqsubseteq A_m$, where $A_l \in \mathcal{T}_l$, $A_m \in \mathcal{T}_m$, A_l and A_m are both atomic concepts. A_m is called the anchor of A_l.*

The key point in the lexical anchoring (or matching) process is to measure how much an atomic concept A_l in a local TBox $\mathcal{T}_l$ is related to an atomic concept A_m in the mediator TBox $\mathcal{T}_m$. This is done by syntactically comparing concepts names (labels). Many lexical similarity measures, proposed in the literature [14], [10], [20], may be used and, as noticed in [20], no similarity measure can give good results in all cases: it is still necessary to look for the best one for each specific application. However, whatever the application is, the relation between A_l and A_m is obtained as follows, considering that φ is the chosen similarity measure: $\Gamma_{N_l} \times \Gamma_{N_m} \rightarrow [0,1]$, where Γ_{N_l}, Γ_{N_m} are respectively the set of atomic concept names in $\mathcal{T}_l$ and $\mathcal{T}_m$. In general, let $n_l \in \Gamma_{N_l}$ be the name of A_l and $n_m \in \Gamma_{N_m}$ the name of A_m, the mapping $m = A_l \sqsubseteq A_m$ is established if and only if $(\varphi(n_l, n_m) \geq \alpha)$ *and* $(\forall n_{mi} \in \Gamma_{N_m} \varphi(n_l, n_m) \geq \varphi(n_l, n_{mi}))$ or $(\alpha \geq \varphi(n_l, n_m) \geq \beta)$ *and* $(n_m \succ n_l)$, where α, β are respectively the maximum and the minimum threshold similarity and $\succ$ denotes a lexical inclusion relation.

Semantic Anchoring Process. It consists in finding additional local concepts that may be relevant for the application domain and which have not been anchored during the lexical anchoring process. We assume that all atomic concepts subsumed by an anchored concept are relevant for the application domain and then must be considered as anchored concepts even if they have not been anchored during the lexical anchoring process. To identify such concepts we automatically compute what we call the *anchoring closure* of a local TBox based on subsumption relationship.

Definition 4.2. *Let $\mathcal{T}_l$ be a local TBox, $\mathcal{T}_m$ the mediator TBox, and $\mathcal{M} = \langle m_1, ..., m_n \rangle$ the result of anchoring $\mathcal{T}_l$ w.r.t $\mathcal{T}_m$. The anchoring closure of $\mathcal{T}_l$, denoted by $a\_cl(\mathcal{T}_l)$, is inductively defined as follows:*

1. *All assertions in $\mathcal{M}$ are also assertions in $a\_cl(\mathcal{T}_l)$.*
2. *All ACI assertions in $\mathcal{T}_l$ are also assertions in $a\_cl(\mathcal{T}_l)$.*
3. *If A_1, A_2, A_3 are atomic concepts and $A_1 \sqsubseteq A_2$ and $A_2 \sqsubseteq A_3$ are in $a\_cl(\mathcal{T}_l)$, then $A_1 \sqsubseteq A_3$ is in $a\_cl(\mathcal{T}_l)$.*

According to this definition we can say that an anchored concept A_l of $\mathcal{T}_l$ is a concept appearing in $a\_cl(\mathcal{T}_l)$ and is of the form: $A_l \sqsubseteq mo : A_m$, where $A_m \in \mathcal{T}_m$ and mo : is a prefix used to distinguish anchor concepts from other concepts.

For example, consider the TBox in Fig. 3(a). It is composed of ACIs of the local TBox $\mathcal{T}_l$, shown graphically in Fig. 2(a), enriched with assertions of the lexical anchoring of $\mathcal{T}_l$ w.r.t $\mathcal{T}_m$. The anchoring closure of this TBox is shown in Fig. 3(b). Notice in Fig. 2(a) that we have *Niambi* that appears as an unanchored concept, because there is no assertion of the form $Niambi \sqsubseteq mo : A_m$, where $A_m \in \mathcal{T}_m$. But, we have the assertion *Niambi* $\sqsubseteq$ *mo:Varieties* in the anchoring closure in Fig. 3(b), and *Varieties* $\in \mathcal{T}_m$. So, *Niambi* becomes an anchored concept, semantically selected, and its anchor is *Varieties*.

ACI assertions	Lexical anchoring assertions
Hotel $\sqsubseteq$ *Accomodation*	*Activity* $\sqsubseteq$ *mo:Activities*
House $\sqsubseteq$ *Accomodation*	*Agriculture* $\sqsubseteq$ *mo:Agriculture*
Tomate $\sqsubseteq$ *CropsVarieties*	*StatDepartement* $\sqsubseteq$ *mo:Departement*
FriedRice $\sqsubseteq$ *CropsVarieties*	*CropsVarieties* $\sqsubseteq$ *mo:Varieties*
Niambi $\sqsubseteq$ *CropsVarieties*	*Tomate* $\sqsubseteq$ *mo:Tomato*
	FriedRice $\sqsubseteq$ *mo:Rice*

(a) ACI assertions with lexical anchoring ones

Anchoring closure assertions	
Activity $\sqsubseteq$ *mo:Activities*	*Tomate* $\sqsubseteq$ *CropsVarieties*
Agriculture $\sqsubseteq$ *mo:Agriculture*	*FriedRice* $\sqsubseteq$ *CropsVarieties*
StatDepartement $\sqsubseteq$ *mo:Departement*	*Niambi* $\sqsubseteq$ *CropsVarieties*
CropsVarieties $\sqsubseteq$ *mo:Varieties*	*Tomate* $\sqsubseteq$ *mo:Varieties*
Tomate $\sqsubseteq$ *mo:Tomato*	*FriedRice* $\sqsubseteq$ *mo:Varieties*
FriedRice $\sqsubseteq$ *mo:Rice*	*Niambi* $\sqsubseteq$ *mo:Varieties*
Hotel $\sqsubseteq$ *Accomodation*	*House* $\sqsubseteq$ *Accomodation*

(b) Anchoring closure TBox

Fig. 3. Semantic anchoring process

From Anchoring to Agreement. We build the agreement of $\mathcal{T}_l$ w.r.t $\mathcal{T}_m$ starting from anchored concepts, *i.e.* those in $a\_cl(\mathcal{T}_l)$. The agreement of $\mathcal{T}_l$ w.r.t $\mathcal{T}_m$ is indeed a TBox $\mathcal{T}_a$ composed by $\mathcal{T}_l'$, a subset of $\mathcal{T}_l$ containing assertions of $\mathcal{T}_l$ that are relevant for the application domain, and $\mathcal{M}$ the result of anchoring $\mathcal{T}_l$ w.r.t $\mathcal{T}_m$. We compute $\mathcal{T}_l'$ by selecting in $\mathcal{T}_l$ assertions that are related to anchored concepts.

Precisely, we aim to select assertions such that: *(i)* $\mathcal{T}_l'$ contains the maximum possible of relevant assertions w.r.t. the application domain, *(ii)* $\mathcal{T}_l'$ contains the minimum possible of irrelevant assertions w.r.t. the application domain, and *(iii)* $\mathcal{T}_l'$ is consistent if $\mathcal{T}_l$ is consistent.

Thus, in addition to anchored concepts, $\mathcal{T}_l'$ may contain unanchored concepts that we call *selected concepts*. A selected concept C is an unanchored concept that must be related to an anchored concept A in order to avoid loosing information about A and also to avoid inconsistency in $\mathcal{T}_l'$. We consider that an unanchored concept C must be a selected concept if $\mathcal{T}_l$ contains assertions of the form:

- $A \sqsubseteq C$, where A is an anchored concept.
- $\exists R \sqsubseteq A$ *and* $\exists R^- \sqsubseteq C$, where A is an anchored concept and R is a basic role.
- $C \sqsubseteq C_1$, where C_1 is a selected concept.

$\exists$ *relateTo* $\sqsubseteq$ *Activity*	$\exists$ *concern* $\sqsubseteq$ *Agriculture*
$\exists$ *relateTo*$^-$ $\sqsubseteq$ *Agriculture*	$\exists$ *concern*$^-$ $\sqsubseteq$ *CropsVarieties*

Fig. 4. Case of selected concept

For instance, consider the local TBox $\mathcal{T}_l$ in Fig. 2 and assume that the concept *Agriculture* is an unanchored concept. Because we have in $\mathcal{T}_l'$ the indirect relation between the two anchored concepts *Activity* and *CropsVarieties*, as illustrated with assertions in Fig. 4, it is necessary to select the concept *Agriculture* in order to keep it in $\mathcal{T}_l'$.

Definition 4.3. *Let $\mathcal{T}_l$ be a local TBox and $\mathcal{T}_m$ be the mediator TBox, the agreement $\mathcal{T}_a = \left\langle \mathcal{T}_l', \mathcal{M} \right\rangle$ for $\mathcal{T}_l$ w.r.t $\mathcal{T}_m$ is such that (i) $\mathcal{M} = \langle m_1, ..., m_n \rangle$ is the result of the anchoring of $\mathcal{T}_l$ w.r.t $\mathcal{T}_m$, and (ii) $\mathcal{T}_l'$ is inductively defined as follows:*

1. *All assertions in $\mathcal{M}$ are in $\mathcal{T}_a$.*
2. *If A is an anchored concept and $B \sqsubseteq A$ is in $\mathcal{T}_l$, then $B \sqsubseteq A$ is in $\mathcal{T}_{l'}$.*
3. *If A is an anchored or a selected concept and $A \sqsubseteq B$ is in $\mathcal{T}_l$, then $A \sqsubseteq B$ is in $\mathcal{T}_l'$.*
4. *If $Q \sqsubseteq R$ is in $\mathcal{T}_l$ and $\exists R \sqsubseteq B$ is in $\mathcal{T}_l'$, then $Q \sqsubseteq R$ is in $\mathcal{T}_l'$.*
5. *If $(funct\ Q)$ is in $\mathcal{T}_l$ and $\exists Q \sqsubseteq B$ is in $\mathcal{T}_l'$, then $(funct\ Q)$ is in $\mathcal{T}_l'$.*
6. *If $\rho(U_C) \sqsubseteq F$ is in $\mathcal{T}_l$ and $B \sqsubseteq \delta(U_C)$ is in $\mathcal{T}_l'$, then $\rho(U_C) \sqsubseteq F$ is in $\mathcal{T}_l'$.*
7. *If $U_C \sqsubseteq V_C$ is in $\mathcal{T}_l$ and $B \sqsubseteq \delta(V_C)$ is in $\mathcal{T}_l'$, then $U_C \sqsubseteq V_C$ is in $\mathcal{T}_l'$.*
8. *If $(funct\ U_C)$ is in $\mathcal{T}_l$ and $\exists Q \sqsubseteq B$ is in $\mathcal{T}_l'$, then $(funct\ U_C)$ is in $\mathcal{T}_l'$.*

Notice that all rules in Definition 4.3 are designed to keep in $\mathcal{T}_l'$ as much semantic information contained in $\mathcal{T}_l$ as possible. Fig 5 shows the agreement TBox computed from assertions of the local TBox $\mathcal{T}_l$ shown graphically in Fig. 2(a). Anchored concepts are those obtained in the example shown in Fig. 3(b).

Local TBox		Agreement TBox	
$\exists$ *manageDataFor* $\sqsubseteq$ *StatDepartement*	$\exists$ *manageDataFor*$^-$ $\sqsubseteq$ *Activity*	*StatDepartement* $\sqsubseteq$ *mo:Departement*	*Activity* $\sqsubseteq$ *mo:Activities*
$\exists$ *relateTo* $\sqsubseteq$ *Activity*	$\exists$ *relateTo*$^-$ $\sqsubseteq$ *Accomodation*	$\exists$ *manageDataFor* $\sqsubseteq$ *StatDepartement*	$\exists$ *manageDataFor*$^-$ $\sqsubseteq$ *Activity*
$\exists$ *relateTo*$^-$ $\sqsubseteq$ *Agriculture*	*Accomodation* $\sqsubseteq$ δ*(capacity)*	*Agriculture* $\sqsubseteq$ *mo:Agriculture*	$\exists$ *relateTo* $\sqsubseteq$ *Activity*
Accomodation $\sqsubseteq$ δ*(rate)*	ρ*(capacity)* $\sqsubseteq$ *xsd:string*	$\exists$ *relateTo*$^-$ $\sqsubseteq$ *Agriculture*	$\exists$ *concern* $\sqsubseteq$ *Agriculture*
ρ*(rate)* $\sqsubseteq$ *xsd:string*	*Hotel* $\sqsubseteq$ *Accomodation*	$\exists$ *concern*$^-$ $\sqsubseteq$ *CropsVarieties*	*CropsVarieties* $\sqsubseteq$ *mo:Varieties*
House $\sqsubseteq$ *Accomodation*	*House* $\sqsubseteq \neg$ *Hotel*	*CropsVarieties* $\sqsubseteq$ δ*(type)*	*CropsVarieties* $\sqsubseteq$ δ*(price_range)*
$\exists$ *concern* $\sqsubseteq$ *Agriculture*	$\exists$ *concern*$^-$ $\sqsubseteq$ *CropsVareities*	ρ*(type)* $\sqsubseteq$ *xsd:string*	ρ*(price_range)* $\sqsubseteq$ *xsd:string*
Cropsvarieties $\sqsubseteq$ δ*(type)*	*Cropsvarieties* $\sqsubseteq$ δ*(price_range)*	*Tomate* $\sqsubseteq$ *mo:Tomato*	*Tomate* $\sqsubseteq$ *Cropsvarieties*
ρ*(type)* $\sqsubseteq$ *xsd:string*	ρ*(price_range)* $\sqsubseteq$ *xsd:string*	*FriedRice* $\sqsubseteq$ *mo:Rice*	*Niambi* $\sqsubseteq$ *Cropsvarieties*
Tomate $\sqsubseteq$ *Cropsvarieties*	*FriedRice* $\sqsubseteq$ *Cropsvarieties*	*Tomate* $\sqsubseteq \neg$ *Niambi*	*Tomate* $\sqsubseteq \neg$ *FriedRice*
Niambi $\sqsubseteq$ *Cropsvarieties*	*Tomate* $\sqsubseteq \neg$ *Niambi*	*FriedRice* $\sqsubseteq \neg$ *Niambi*	
Tomate $\sqsubseteq \neg$ *FriedRice*	*FriedRice* $\sqsubseteq \neg$ *Niambi*		

Fig. 5. Example of an agreement TBox

5 Conciliation

We can now build the global TBox $\mathcal{T}_g$ by conciliating the different agreement TBoxes $\mathcal{T}_{ai} = \left\langle \mathcal{T}_{li}', \mathcal{M}_i \right\rangle$ obtained above. The conciliation is achieved incrementally by integrating the agreement TBoxes into $\mathcal{T}_g$, one after another. Integrating an agreement TBox $\mathcal{T}_a$ in $\mathcal{T}_g$ consists in linking its concepts with the ones of other agreement TBoxes already conciliated in $\mathcal{T}_g$. Links between concepts in $\mathcal{T}_g$ are established through anchor concepts contained in $\mathcal{M}_i$ for every agreement TBox $\mathcal{T}_{ai}$. Let us recall that all anchor concepts are part of the mediator TBox $\mathcal{T}_m$. Thus, we search for links between anchor concepts in $\mathcal{T}_m$ in order to use them to conciliate concepts in $\mathcal{T}_g$. In this way, our global TBox $\mathcal{T}_g$ contains the following components:

- the set of agreement TBoxes $\mathcal{T}_{ai} = \langle \mathcal{T}'_{li}, \mathcal{M}_i \rangle$. They represent the part of local TBoxes that are shared ($\mathcal{T}'_{li}$), together with the mappings between these local concepts and the mediator ones ($\mathcal{M}_i$).
- an as small as possible subset $\mathcal{T}'_m$ of $\mathcal{T}_m$ containing only the part of the hierarchy which is usefull to link local concepts.

To illustrate this process in the context of agricultural domain, consider the example in Fig. 6. In this example the concepts *Tomate* of the agreement TBox $\mathcal{T}_{a1}$ and *FriedRice* of the agreement TBox $\mathcal{T}_{a2}$ are respectively anchored by the concepts *Tomato* and *Rice* of the mediator TBox $\mathcal{T}_m$. The structure of the mediator TBox reveals that *Tomato* and *Rice* have a common ancestor which is the concept *plan_product*. We reproduce this relation to conciliate the concepts *Tomate* and *FriedRice* in the global TBox $\mathcal{T}_g$.

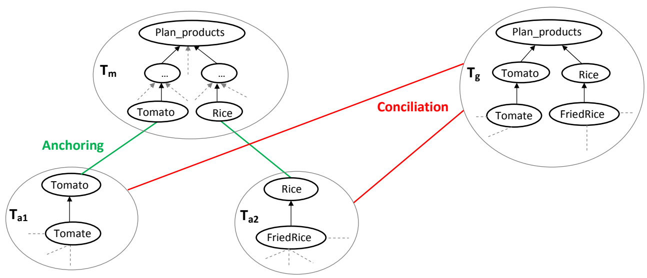


Fig. 6. General overview of the conciliation phase

Definition 5.1. *Let $\{\mathcal{T}_{li}\}$ be a set of local TBoxes and $\mathcal{T}_m$ be a mediator TBox. The corresponding Global TBox $\mathcal{T}_g$ is $\langle \{\mathcal{T}_{ai}\}, \mathcal{T}'_m \rangle$, where (i) $\{\mathcal{T}_{ai}\}$ is the set of agreement TBoxes built from local TBoxes according to Definition 4.3, and (ii) $\mathcal{T}'_m$ is the smallest subset of $\mathcal{T}_m$ that conciliates every $\mathcal{T}_{ai}$ in $\mathcal{T}_g$, built by Algorithm 1.*

As suggested by the example in Fig. 6, one particular interest in our approach is the use of the hierarchy of the mediator TBox $\mathcal{T}_m$ in order to find links between anchor concepts. These links are reproduced in the global TBox for conciliating agreements.

The relation that we are looking for within the hierarchy of $\mathcal{T}_m$ is the *least common subsumer* (*lcs*) of two anchor concepts. It is important to notice that in our first experiments we have only considered tree taxonomies, we plan to generalize this point in future work. We can follow the algorithm proposed in [5] to compute the *lcs* of two concepts C_1 and C_2 in $\mathcal{T}_m$, according to the definition of *lcs* that we recall hereafter.

Definition 5.2. *Let $\mathcal{T}_m$ be the mediator TBox, C_1 and C_2 two given atomic concepts in $\mathcal{T}_m$, the concept C of $\mathcal{T}_m$ is the lcs of C_1 and C_2 in $\mathcal{T}_m$ ($C = lcs_{\mathcal{T}_m}(C_1, C_2)$) iff (i) $C_i \sqsubseteq C$ for $i = 1,2$, and (ii) C is the least concept with this property, i.e. if C' satisfies $C_i \sqsubseteq C'$ for $i = 1,2$, then $C \sqsubseteq C'$.*

Based on *lcs* computation in [5], $\mathcal{T}'_m$ consists in a subsumption hierarchy between all anchor concepts of all $\mathcal{T}_{ai}$ and their *lcs* in $\mathcal{T}_m$. The algorithm that we propose to achieve this uses the hierarchical proximity measure proposed by [22], that we recall in the following definition.

Definition 5.3. *Let $\mathcal{T}_m$ be the mediator TBox, C_1 and C_2 two concepts of $\mathcal{T}_m$. The hierarchical proximity measure between C_1 and C_2 in $\mathcal{T}_m$ is such that:*

$$sim_{\mathcal{H}}(C_1, C_2) = \frac{2 * depthOf(lcs_{\mathcal{T}_m}(C_1, C_2))}{depthOf(C_1) + depthOf(C_2)},$$

where $depthOf(C)$ returns the number of subsumers of C in $\mathcal{T}_m$.

Definition 5.4. *Let $C \in \mathcal{T}_m$. If $sim_{\mathcal{H}}(C, C_j) \geq sim_{\mathcal{H}}(C, C_i), \forall C_i \in \mathcal{T}_m$, then we say that C_j is the closest concept of C in $\mathcal{T}_m$ and we denote it by $closest_{\mathcal{T}_m}(C)$.*

If $sim_{\mathcal{H}}(C, C_j) \geq sim_{\mathcal{H}}(C, C_i), \forall C_i \in \mathcal{T}'_m \subseteq \mathcal{T}_m$, then we say that C_j is the closest concept of C in $\mathcal{T}'_m$ w.r.t. $\mathcal{T}_m$ and we denote it by $closest_{\mathcal{T}'_m/\mathcal{T}_m}(C)$.

The conciliation of an agreement TBox $\mathcal{T}_{ak} = \langle \mathcal{T}'_{lk}, \mathcal{M}_k \rangle$ with others agreement TBoxes already conciliated in $\mathcal{T}_g = \langle \{\mathcal{T}_{ai}\}_{i \neq k}, \mathcal{T}'_m \rangle$ consists in integrating each anchor concept A_m of $\mathcal{M}_k$ within the hierarchy $\mathcal{T}'_m$. To integrate a concept A_m within the hierarchy $\mathcal{T}'_m$ we have to compute the *lcs* in $\mathcal{T}_m$ between A_m and the closest concept of A_m in $\mathcal{T}_m$ among the anchor concepts already present in the hierarchy $\mathcal{T}'_m$. In order to express these features in our conciliation algorithm, we use Definitions 5.4, as it can be noticed in what follows:

Algorithm 1. ***Conciliation***

Input: $\mathcal{T}_{ak} = \langle \mathcal{T}'_{lk}, \mathcal{M}_k \rangle$, $\mathcal{T}_g = \langle \{\mathcal{T}_{ai}\}_{i \neq k}, \mathcal{T}'_m \rangle$
Output: $\mathcal{T}_g = \langle \{\mathcal{T}_{ai}\} \cup \mathcal{T}_{ak}, \mathcal{T}'_m \rangle$

for each *($m_k = A_l \sqsubseteq A_m$ in $\mathcal{M}_k$)* ***do***
    if *(($\mathcal{T}'_m \neq \emptyset$) and ($A_m \notin \mathcal{T}'_m$))* ***then***
        $A_{cl} \leftarrow closest_{\mathcal{T}'_m/\mathcal{T}_m}(A_m)$
        $A_{lcs} \leftarrow lcs_{\mathcal{T}_m}(A_m, A_{cl})$
        if *($A_{lcs} = A_{cl}$)* ***then***
            $\mathcal{T}'_m \leftarrow \mathcal{T}'_m \cup \{A_m \sqsubseteq A_{cl}\}$
        else if *($A_{lcs} = A_m$)* ***then***
            $\mathcal{T}'_m \leftarrow \mathcal{T}'_m \cup \{A_{cl} \sqsubseteq A_m\}$
            if *($\exists A \in \mathcal{T}'_m \mid A = lcs_{\mathcal{T}'_m}(A_{cl}, A)$)* ***then***
                $\mathcal{T}'_m \leftarrow \mathcal{T}'_m \cup \{A_m \sqsubseteq A\}$
            end if
        else
            $\mathcal{T}'_m \leftarrow \mathcal{T}'_m \cup \{A_m \sqsubseteq A_{lcs}, A_{cl} \sqsubseteq A_{lcs}\}$
            if *($\exists A \in \mathcal{T}'_m \mid = lcs_{\mathcal{T}'_m}(A_{cl}, A)$)* ***then***
                $\mathcal{T}'_m \leftarrow \mathcal{T}'_m \cup \{A_m \sqsubseteq A\}$
            end if
        end if
    else if *($\mathcal{T}'_m = \emptyset$)* ***then***
        $\mathcal{T}'_m \leftarrow \mathcal{T}'_m \cup \{A_m \sqsubseteq \top\}$
    end if
end for

To illustrate our algorithm, we consider two TBoxes $\mathcal{T}_{a1} = \left\langle \mathcal{T}'_{l1}, \mathcal{M}_1 \right\rangle$ and $\mathcal{T}_{a2} = \left\langle \mathcal{T}'_{l2}, \mathcal{M}_2 \right\rangle$ such that :

- $\mathcal{M}_1 = \langle FrideRice \sqsubseteq mo : rice, Onion \sqsubseteq mo : Onion \rangle$
- $\mathcal{M}_2 = \langle Sorgho \sqsubseteq mo : Sorgho, Tomate \sqsubseteq mo : Tomato \rangle$

Results obtained by conciliating $\mathcal{T}_{a1}$ and $\mathcal{T}_{a2}$ are as follows:

1- **Integrate $\mathcal{T}_{a1}$ in $\mathcal{T}_g$**
Input: $\mathcal{T}_{a1}$, $\mathcal{T}_g = \langle \{\}, \mathcal{T}'_m = \emptyset \rangle$

iteration 1 $- \; m_{1.1} = FrideRice \sqsubseteq mo : Rice$
$\mathcal{T}'_m = \{Rice \sqsubseteq \top\}$
iteration 2 $- \; m_{1.2} = Onion \sqsubseteq mo : Onion$
$A_{cl} = Rice$; $A_{lcs} = PlanProducts$
$\mathcal{T}'_m = \{Rice \sqsubseteq PlanProducts, Onion \sqsubseteq PlanProducts\}$

2- **conciliate $\mathcal{T}_{a1}$ and $\mathcal{T}_{a2}$ in $\mathcal{T}_g$**
Input: $\mathcal{T}_{a2}$, $\mathcal{T}_g = \langle \{\mathcal{T}_{a1}\}, \mathcal{T}'_m = \{Rice \sqsubseteq PlanProducts, Onion \sqsubseteq PlanProducts\} \rangle$

iteration 1 $- \; m_{2.1} = Sorgho \sqsubseteq mo : Sorgho$
$A_{cl} = Rice; A_{lcs} = Cereals$
$\mathcal{T}'_m = \{Rice \sqsubseteq Cereals, Sorgho \sqsubseteq Cereals, Cereals \sqsubseteq PlanProducts, Onion \sqsubseteq PlanProducts\}$
iteration 2 $- \; m_{2.2} = Tomate \sqsubseteq mo : Tomato$
$A_{cl} = Onion$; $A_{lcs} = Vegetables$
$\mathcal{T}'_m = \{Rice \sqsubseteq Cereals, Sorgho \sqsubseteq Cereals, Cereals \sqsubseteq PlanProducts, Onion \sqsubseteq Vegetables, Tomato \sqsubseteq Vegetables, Vegetables \sqsubseteq PlanProducts\}$

Fig. 7 illustrates graphically the global TBox $\mathcal{T}_g$ resulting from the conciliation of $\mathcal{T}_{a_1}$ and $\mathcal{T}_{a_2}$. We have distinguished the hierarchy $\mathcal{T}'_m$, composed of all anchor concepts in $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$, linked to each other by containment assertions found in $\mathcal{T}_m$. Notice that all information existing in local TBoxes also exists in $\mathcal{T}_g$. In fact, the part of $\mathcal{T}_g$ which is not in $\mathcal{T}'_m$ represents the data sources that can be accessed from the global schema, this access being supported by the mapping $\mathcal{M}$ (following a GAV approach).

It can be noticed that the global TBox $\mathcal{T}_g$ is composed of *(i)* the union of all $\mathcal{T}'_l$ (local parts of each source agreement), *(ii)* the union of all $\mathcal{M}$ (mapping parts of each source agreement) and *(iii)* the hierarchy extracted from $\mathcal{T}_m$ to relate anchors in $\mathcal{T}_g$. Thus, managing dynamic changes that occur frequently in a semantic web context requires to consider not only to add new sources but also to remove a source or to update a source's schema. We consider these three operations: adding a new source S consists of the following steps:

– the computation of S's agreement: $\mathcal{T}_a = \left\langle \mathcal{T}'_l, \mathcal{M} \right\rangle$,
– the union of $\mathcal{T}'_l$ with local parts of the other sources,
– the union of $\mathcal{M}$ with mapping parts of the other sources,
– the integration of anchors of $\mathcal{M}$ into the hierarchy $\mathcal{T}'_m$ (using Algorithm 1).

Removing a source S that became unavailable will follow two stages:

- the removal from $\mathcal{T}_g$ of $\mathcal{T}_a = \langle \mathcal{T}_l', \mathcal{M} \rangle$ corresponding to S,
- an iteration on the hierarchy $\mathcal{T}_m'$, in order to remove items that became unnecessary. We plan to design the corresponding algorithm in future works and to compare it with the simple recomputation of $\mathcal{T}_m'$ based on remaining sources.

Finally, taking into account an update performed on a source TBox $\mathcal{T}_l$ will require first to compute the corresponding new agreement. This is again one of our future works to design an incremental update algorithm, more efficient than removing the old version and adding the new one.

6 Conclusion and Future Work

Our proposition in this article brings a solution to the problem of *automatic* construction of an appropriate global ontology. This ontology will be shared between a set of loosely-interrelated partners. We have tackled this problem using a background-knowledge, i.e., a domain-reference ontology, as a mediator to build the global ontology. The global ontology offers interesting properties, especially an appropriate conceptualization and easy resource-adding and querying processes. For our solution to be automated, we use, on one hand, logical-inference techniques offered by description logics, the knowledge-representation formalism used for ontology specification. On the other hand, we make use of some classical syntactic-matching techniques for ontology matching. Our approach is that hybrid. To the best of our knowledge, no other solution has already been proposed in the literature combining these two techniques for an automatic construction of a global ontology.

We are working to go further in exploring automatic-reasoning capabilities of *DL-Lite*$_{\mathcal{A}}$ Description Logic, in order to (i) check the global-ontology's consistency and (ii) answer queries using the global ontology. Moreover, in future work all types of relations will be considered in the reference ontology, we will thus extend our proposition

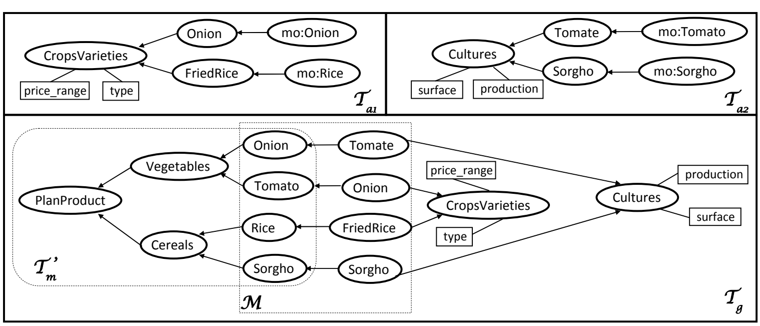


Fig. 7. Overview of a global TBox that conciliates two agreement TBoxes

that concerns here only subsumption relationship. Another important future work is to specify the complete life cycle management of the global ontology.

Finally, we will build upon our first experiments, which have been realized as a proof of concept but are not yet an actual publishable evaluation, in order to turn our proposition into a robust software for ontology-based data integration.

References

1. Aleksovski, Z., Klein, M.C.A., ten Kate, W., van Harmelen, F.: Matching Unstructured Vocabularies Using a Background Ontology. In: Staab, S., Svátek, V. (eds.) EKAW 2006. LNCS (LNAI), vol. 4248, pp. 182–197. Springer, Heidelberg (2006)
2. Aleksovski, Z., ten Kate, W., van Harmelen, F.: Exploiting the Structure of Background Knowledge Used in Ontology Matching. In: Ontology Matching (2006)
3. Amann, B., Beeri, C., Fundulaki, I., Scholl, M.: Ontology-based integration of XML web resources. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 117–131. Springer, Heidelberg (2002)
4. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, Cambridge (2003)
5. Baader, F., Sertkaya, B., Turhan, A.-Y.: Computing the least common subsumer w.r.t. a background terminology. J. Applied Logic 5(3), 392–420 (2007)
6. Bergamaschi, S., Castano, S., Vincini, M., Beneventano, D.: Semantic integration of heterogeneous information sources. Data Knowl. Eng. 36(3), 215–249 (2001)
7. Borgida, A., Serafini, L.: Distributed description logics: Directed domain correspondences in federated information sources. In: Chung, S., et al. (eds.) CoopIS 2002, DOA 2002, and ODBASE 2002. LNCS, vol. 2519, pp. 36–53. Springer, Heidelberg (2002)
8. Borgida, A., Serafini, L.: Distributed description logics: Assimilating information from peer sources. In: Spaccapietra, S., March, S., Aberer, K. (eds.) Journal on Data Semantics I. LNCS, vol. 2800, pp. 153–184. Springer, Heidelberg (2003)
9. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable Reasoning and Efficient Query Answering in Description Logics: The *L-Lite* Family. J. Autom. Reasoning 39(3), 385–429 (2007)
10. Choi, N., Song, I.-Y., Han, H.: A survey on ontology mapping. SIGMOD Record 35(3), 34–41 (2006)
11. Cruz, I.F., Xiao, H.: Ontology driven data integration in heterogeneous networks. In: Complex Systems in Knowledge-based Environments, pp. 75–98 (2009)
12. Cruz, I.F., Xiao, H., Hsu, F.: Peer-to-peer semantic integration of XML and RDF data sources. In: Moro, G., Bergamaschi, S., Aberer, K. (eds.) AP2PC 2004. LNCS (LNAI), vol. 3601, pp. 108–119. Springer, Heidelberg (2005)
13. Grau, B.C., Parsia, B., Sirin, E.: Combining owl ontologies using epsilon-connections. J. Web Sem. 4(1), 40–59 (2006)
14. Kalfoglou, Y., Schorlemmer, M.: Ontology Mapping: The State of the Art. The Knowledge Engineering Review 18 (2003)
15. Lenzerini, M.: Data integration: A theoretical perspective. In: PODS, pp. 233–246 (2002)
16. Mena, E., Illarramendi, A., Kashyap, V., Sheth, A.P.: OBSERVER: An Approach for Query Processing in Global Information Systems Based on Interoperation Across Pre-Existing Ontologies. Distributed and Parallel Databases 8(2), 223–271 (2000)
17. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking Data to Ontologies. In: Spaccapietra, S. (ed.) Journal on Data Semantics X. LNCS, vol. 4900, pp. 133–173. Springer, Heidelberg (2008)

18. Sabou, M., d'Aquin, M., Motta, E.: Using the Semantic Web as Background Knowledge for Ontology Mapping. In: Ontology Matching (2006)
19. Serafini, L., Borgida, A., Tamilin, A.: Aspects of distributed and modular ontology reasoning. In: IJCAI, pp. 570–575 (2005)
20. Shvaiko, P., Euzenat, J.: Ten Challenges For Ontology Matching. In: Proceedings of The 7th International Conference on Ontologies, DataBases, and Applications of Semantics, ODBASE (2008)
21. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hübner, S.: Ontology-Based Integration of Information - A Survey of Existing Approaches. In: IJCAI 2001 Workshop on Ontologies and Informations Sharing, pp. 108–117 (2001)
22. Wu, Z., Palmer, M.S.: Verb Semantics and Lexical Selection. In: ACL, pp. 133–138 (1994)