Modeling Ontologies Using OWL, Description Graphs, and Rules

Boris Motik[1], Bernardo Cuenca Grau[1], Ian Horrocks[1], and Ulrike Sattler[2]

[1] University of Oxford, UK
[2] University of Manchester, UK

1 Introduction

Ontologies often describe *structured objects*, which consist of many parts interconnected in complex ways. Such objects abound in molecular biology and the clinical sciences. Clinical ontologies such as GALEN, the Foundational Model of Anatomy (FMA), and the National Cancer Institute (NCI) Thesaurus describe numerous structured objects. For example, FMA models the human hand as consisting of the fingers, the palm, various bones, blood vessels, and so on, all of which are highly interconnected.

Modeling structured objects poses numerous problems to the OWL family of languages. The design of OWL DL and OWL 2 has been driven by the desire to provide practically useful knowledge modeling primitives while ensuring decidability of reasoning. The latter goal has been achieved by ensuring that the selected primitives have a variant of the *tree-model property* [1]: each satisfiable OWL knowledge base has a model whose elements are connected in a tree-like manner. Thus, OWL ontologies describing (usually non-tree-like) structured objects typically have models that do not reflect the actual structure of the modeled objects. This technical problem has severe consequences for OWL users [2]. In search of the "correct" way of describing structured objects, modelers often create overly complex ontologies; however, since the required expressive power is actually missing, these ontologies do not entail the consequences that would follow if the objects were described accurately. Furthermore, the complexity of the ontologies can cause significant performance problems during reasoning.

To address this lack of expressivity, we propose to extend OWL with *description graphs*, which can be understood as schema-level descriptions of structured objects. Furthermore, to allow the representation of conditional statements about structured objects, we also extend OWL with first-order rules [3]. For example, we can represent the structure of the hand and its parts using description graphs, and we can represent statements such as "if a bone in the hand is fractured, then the hand is fractured as well" using rules. Finally, we can use standard OWL-style modeling to represent nonstructural aspects of the domain, such as "a medical doctor is a person with an MD degree."

We thus obtain a powerful knowledge representation formalism that addresses the expressivity limitations of OWL, but that is, unfortunately, undecidable. It is widely recognized that reasoning algorithms are more likely to be effective in

practice if the underlying logics are decidable. Therefore, we have analyzed the main causes for undecidability and have investigated restrictions under which the formalism becomes decidable.

In particular, we have observed that structured objects can often be described by a possibly large, yet bounded number of parts. For example, a human body consists of a certain number of organs, all of which can be decomposed into smaller parts; further decomposition, however, will eventually reach the parts that the modeler cannot or does not want to describe. For example, FMA describes the skeleton of the hand, but it does not describe the structure of the distal phalanges of the fingers. The number of parts needed to describe the hand is thus determined by the granularity of the hierarchical decomposition of the hand. This decomposition naturally defines an acyclic hierarchy of description graphs. For example, the fingers will be described by description graphs that are subordinate to that of the hand; furthermore, the description graph for the hand is not naturally subordinate to the description graphs for the fingers. We use this observation to define a particular *acyclicity* restriction on description graphs. Roughly speaking, it allows an instance of a description graph up the hierarchy to imply existence of an instance of a description graph lower in the hierarchy, but not vice versa. Provided that the OWL TBox is empty, acyclicity bounds the number of parts that one needs to reason with, which can be used to obtain a decision procedure for the basic reasoning problems.

If the OWL TBox is not empty, the acyclicity condition alone does not ensure decidability due to an interaction between OWL axioms, graphs, and rules [4]. To obtain decidability, we limit their interaction by imposing an additional *role separation* condition. In particular, we separate the roles that can be used in OWL axioms from the roles that can be used in the rules; furthermore, depending on the expressivity of the used fragment of OWL, we may additionally require that the OWL axioms do not refer to the roles used in the description graphs.

This paper summarizes the results published in several recent papers [5, 2]. For the sake of brevity, we omit the proofs and certain technical details, which can be found in [5, 2]. Furthermore, we assume the reader to be familiar with OWL and the basics of description logics (DLs).

2 Problems with Modeling Complex Structures

To understand the limitations of modeling structured objects in DLs (and hence in OWL), we consider the problem of modeling the skeleton of the human hand (see Figure 1a). The carpal bones form the base of the hand. The central part contains the metacarpal bones, one leading to each finger. The fingers consist of phalanges: the proximal phalanges are connected to the metacarpal bones, and all fingers apart from the thumb contain a middle phalanx between the proximal and the distal phalanx. This structure can be conceptualized as in Figures 1b–1e.

Figures 1b–1e could be represented in DLs using an ABox $\mathcal{A}$. ABox assertions, however, represent concrete data; thus, $\mathcal{A}$ would represent the structure of *one particular* hand. In this paper, we are concerned with modeling structured

objects *at the schema level*—that is, we want to describe the general structure of *all* hands. and instantiate such a description many times. For example, if we say that each patient has a hand, then, for each concrete patient, we should instantiate a *different* hand, each of the structure shown in Figures 1b–1e. This cannot be achieved using ABox assertions.

We can give a logical, schema-level interpretation to Figures 1b–1e by treating vertices as concepts and arrows as *participation constraints* specifying their relationships. For example, *Hand* and *Index_finger* are concepts and the arrow between them says that the index is a part of the hand.[3] Participation constraints are represented in ontologies using DL axioms such as (1)–(5).

Let $\mathcal{K}$ be a DL knowledge base containing the following axioms, in which, for the sake of brevity, we omit the suffix *_of_index_finger*.

$$\mathit{Index\_finger} \sqsubseteq \exists \mathit{part}.\mathit{Distal\_phalanx} \tag{1}$$

$$\mathit{Index\_finger} \sqsubseteq \exists \mathit{part}.\mathit{Middle\_phalanx} \tag{2}$$

$$\mathit{Distal\_phalanx} \sqsubseteq \exists \mathit{attached\_to}.\mathit{Middle\_phalanx} \tag{3}$$

$$\mathit{Middle\_phalanx} \sqsubseteq \exists \mathit{attached\_to}.\mathit{Proximal\_phalanx} \tag{4}$$

$$\mathit{Proximal\_phalanx} \sqsubseteq \exists \mathit{part}^{-}.\mathit{Index\_finger} \tag{5}$$

$$\mathsf{Sym}(\mathit{attached\_to}) \tag{6}$$

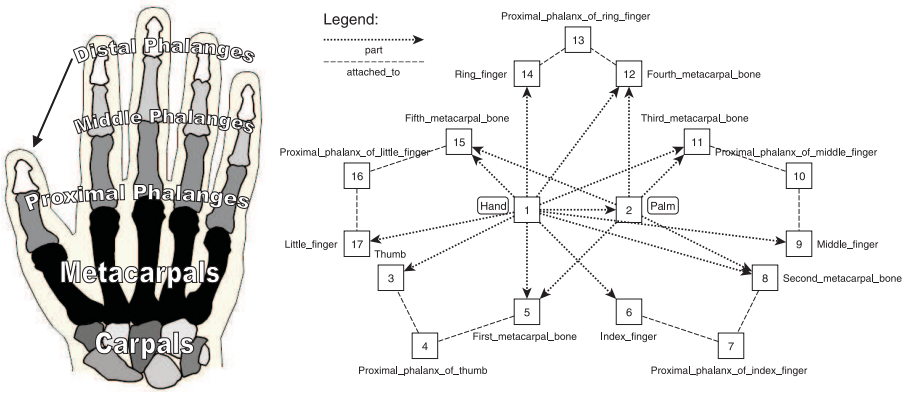
Let I be an interpretation corresponding to Figure 1e in the obvious way. Clearly, I satisfies $\mathcal{K}$, which justifies the formalization of Figure 1e using $\mathcal{K}$. Unfortunately, the ontology $\mathcal{K}$ is underconstrained: some models of $\mathcal{K}$ do not correspond to the actual structure of the index finger from Figure 1e. Axioms (2) and (4) imply the existence of two middle phalanges of the index finger, but $\mathcal{K}$ does not state that these two middle phalanges must be the same object. Thus, the interpretation I' corresponding to Figure 2 is also a model of $\mathcal{K}$.

This discrepancy prevents us from drawing conclusions that rely on the non-tree connections in the structure; for example, if the index finger has a broken distal phalanx, then we should conclude that the phalanx adjacent to the middle phalanx is broken (since this is the same broken phalanx). Furthermore, it can also cause problems with the performance of reasoning. For example, we might use axioms (2)–(6) to describe the relationships between the index finger, its proximal phalanx and its middle phalanx.

The axioms in $\mathcal{K}$ do not state that the index finger in (5) is a part of the "initial" index finger, so the interpretation I' contains an infinite tree of index fingers. In fact, this model is "canonical" in the sense that it reflects the least amount of information derivable from the axioms. In order to disprove an entailment, a DL reasoner will try to construct such a "canonical" model. In practice, these models can be highly repetitive and much larger than the intended ones, which, according to our experience, is the main reason why DL reasoners cannot process ontologies such as FMA and certain versions of GALEN.

These problems could be addressed by ensuring that *all* models of $\mathcal{K}$ resemble as much as possible the intended conceptualization shown in Figures 1b–1e. DL

[3] The role *attached_to* is symmetric, so we do not orient the edges labeled with it.



(a) Anatomy of the Hand (b) Hand (G_{hand})

(c) Finger (G_{finger}) (d) Thumb (G_{thumb}) (e) Index (G_{index_finger})

Fig. 1: The Anatomy of the Hand and its Conceptual Models

languages, however, exhibit (a variant of) the *tree model property* [1]: whenever a DL knowledge base $\mathcal{K}$ has a model, it has a model of a certain tree shape. This is a consequence of the form of axioms allowed in OWL, and is generally considered desirable because it ensures decidability of reasoning. At the same time, however, it also means that we must leave the confines of DLs and OWL if we want to faithfully represent structured objects.

3 The Formalism

We now present our formalism. We first introduce description graphs.

Definition 1 (Description Graph). *An ℓ-ary* description graph *is a directed labeled graph $G = (V, E, \lambda, M)$ with $V = \{1, \ldots, \ell\}$ a set of* vertices*, $E \subseteq V \times V$ a set of* edges*, and λ a labeling function that assigns a set of atomic concepts or the negation of atomic concepts $\lambda\langle i\rangle$ to each vertex $i \in V$ and a set of atomic roles $\lambda\langle i, j\rangle \subseteq N_R$ to each edge $\langle i, j\rangle \in E$. Finally, $M \subseteq N_C$ is a set of* main concepts *for G. For A an atomic concept, V_A is the set of vertices that contain A in their label; that is, $V_A = \{k \in V \mid A \in \lambda\langle k\rangle\}$.*

Thus, description graphs are labeled graphs where the nodes are labeled with concepts and the edges with roles. The main concepts indicate the objects whose

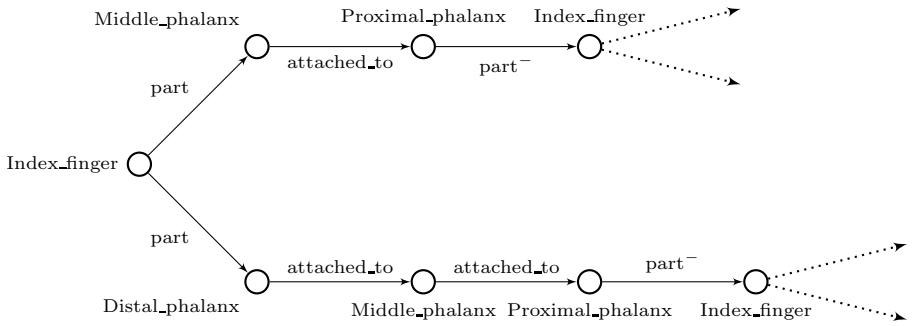


Fig. 2: Unintended Infinite Model I'

structure is defined by the graphs. For example, the main concepts for the graph in Figure 1b (framed with rounded rectangles) are *Hand* and *Palm*, meaning that this graph defines the structure of the hand and the palm. Intuitively, an instance of a main concept implies the existence of a graph instance.

Definition 2 (Rules). *Let N_I and N_V be disjoint sets of individuals and variables. An* atom *is of the form $C(s)$, $R(s,t)$, or $s \approx t$, for $s,t \in N_I \cup N_V$, C a concept, and R a role. A* rule *is an expression of the form*

$$U_1 \wedge \ldots \wedge U_m \rightarrow V_1 \vee \ldots \vee V_n \tag{7}$$

where U_i and V_j are atoms, $m \geq 0$, and $n \geq 0$. W.l.o.g we assume that the body never contains $\approx$. The conjunction $U_1 \wedge \ldots \wedge U_m$ is called the body*, and the disjunction $V_1 \vee \ldots \vee V_n$ is called the* head*. Variables x and y are* directly connected *in a rule r if they both occur in a body atom of r, and* connected *is the transitive closure of directly connected. A rule r is* connected *if each pair of variables x and y occurring in r is connected in r.*

A graph rule *is a rule of the form* (7) *where all concepts and roles in atoms are atomic, and that can also contain* graph atoms *of the form $G(t_1, \ldots, t_k)$, for G an ℓ-ary description graph and $t_i \in N_I \cup N_V$.*

Next, we introduce graph specializations, which allow us to represent objects at different levels of abstraction. For example, we would like to describe the abstract structure common to all fingers as shown in Figure 1c; then, we should be able to specialize this structure for the index finger and introduce the middle phalanx, as in Figure 1e. The graph specialization $G_{finger} \lhd G_{thumb}$ states that the graph for the thumb specializes the graph for a finger.

Definition 3 (Graph Specialization). *A* graph specialization *is an axiom of the form $G_1 \lhd G_2$, where $G_1 = (V_1, E_1, \lambda_1, M_1)$ and $G_2 = (V_2, E_2, \lambda_2, M_2)$ are description graphs with $V_1 \subseteq V_2$.*

Next, we introduce axioms that allow us to properly connect graph instances. For example, G_{hand} contains the vertices 3 and 4 for the thumb and its proximal

phalanx, which correspond to the vertices 1 and 3 of G_{thumb}. We can specify this correspondence using a *graph alignment* of the form $G_{hand}[3,4] \leftrightarrow G_{thumb}[1,3]$. Intuitively, this ensures that it is not possible for G_{hand} and G_{thumb} to share the thumb without sharing the proximal phalanx as well.

Definition 4 (Graph Alignment). *A* graph alignment *is an expression of the form $G_1[v_1, \ldots, v_n] \leftrightarrow G_2[w_1, \ldots w_n]$, where G_1 and G_2 are description graphs with sets of vertices V_1 and V_2, respectively, $v_i \in V_1$ and $w_i \in V_2$ for $1 \leq i \leq n$.*

Finally, we define GBoxes and graph-extended KBs.

Definition 5 (Formalism). *A* graph box *(GBox) is a tuple $\mathcal{G} = (\mathcal{G}_G, \mathcal{G}_S, \mathcal{G}_A)$ where $\mathcal{G}_G$, $\mathcal{G}_S$, and $\mathcal{G}_A$ are finite sets of description graphs, graph specializations over $\mathcal{G}_G$, and graph alignments over $\mathcal{G}_G$. ABoxes are extended to allow for* graph assertions *of the form $G(a_1, \ldots, a_\ell)$ for G an ℓ-ary graph. A* graph-extended knowledge base *is a 4-tuple $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ where $\mathcal{T}$ is a TBox, $\mathcal{P}$ is a program with a finite number of connected rules, $\mathcal{G}$ is a GBox, and $\mathcal{A}$ is an ABox.*

Next, we define the semantics of the formalism.

Definition 6 (Semantics). *An interpretation $I = (\triangle^I, \cdot^I)$ is defined as usual, and it interprets each ℓ-ary description graph G as an ℓ-ary relation over $\triangle^I$; that is, $G^I \subseteq (\triangle^I)^\ell$. A graph assertion is satisfied in I, written $I \models G(a_1, \ldots, a_\ell)$, iff $\langle a_1^I, \ldots, a_\ell^I \rangle \in G^I$. Satisfaction of a description graph, graph specialization, and graph alignment in I is defined in Table 1, and satisfaction of $\mathcal{T}$, $\mathcal{P}$, and $\mathcal{A}$ in I is defined as usual. A knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ is satisfied in I, written $I \models \mathcal{K}$, if all its components are satisfied in I.*

Thus, each ℓ-ary graph G is interpreted as an ℓ-ary relation G^I in which each tuple corresponds to an instance of G in the interpretation. The key and disjointness properties in Table 1 ensure that no two distinct instances of G can share a vertex; for example, no two distinct instances of G_{hand} can share the vertex for the thumb. These properties ensure that no model I contains infinite "chains" of instances of a description graph, which reflects the intuition that each description graph instance represents a bounded and isolated part of the domain. The start property in Table 1 ensures that each instance of a main concept A of G occurs in an instance of G. For example, since *Hand* is a main concept for G_{hand}, each instance of *Hand* must occur as vertex 1 in an instance of G_{hand}.

Graph specializations are interpreted as inclusions over the graph relations; for example, $G_{finger} \lhd G_{index\_finger}$ means that each instance of an index finger is also an instance of a finger. The two graphs share all the vertices of the more general graph, and the more specific graph can introduce additional vertices. Finally, graph alignments state that, whenever two graphs share some vertex from the specified list, then they share all other vertices from the list as well. For example, the alignment $G_{hand}[3,4] \leftrightarrow G_{thumb}[1,3]$ states that, if instances of G_{hand} and G_{thumb} share vertices 3 and 1, respectively, then they must also share vertices 4 and 3, respectively.

Table 1: Satisfaction of GBox Elements in an Interpretation

$I \models G$ for $G = (V, E, \lambda, M)$ iff
Key property:
$\forall x_1, \ldots, x_\ell, y_1, \ldots, y_\ell \in \triangle^I$:
$\langle x_1, \ldots, x_\ell \rangle \in G^I \wedge \langle y_1, \ldots, y_\ell \rangle \in G^I \wedge \bigvee_{1 \leq i \leq \ell} x_i = y_i \rightarrow \bigwedge_{1 \leq j \leq \ell} x_j = y_j$
Disjointness property:
$\forall x_1, \ldots, x_\ell, y_1, \ldots, y_\ell \in \triangle^I : \langle x_1, \ldots, x_\ell \rangle \in G^I \wedge \langle y_1, \ldots, y_\ell \rangle \in G^I \rightarrow \bigwedge_{1 \leq i < j \leq \ell} x_i \neq y_j$
Start property: for each atomic concept $A \in M$,
$\forall x \in \triangle^I : x \in A^I \rightarrow \exists x_1, \ldots, x_\ell \in \triangle^I : \langle x_1, \ldots, x_\ell \rangle \in G^I \wedge \bigvee_{k \in V_A} x = x_k$
Layout property:
$\forall x_1, \ldots, x_\ell \in \triangle^I : \langle x_1, \ldots, x_\ell \rangle \in G^I \rightarrow$
$\bigwedge_{i \in V,\ B \in \lambda\langle i \rangle} x_i \in B^I \wedge \bigwedge_{\langle i,j \rangle \in E,\ R \in \lambda\langle i,j \rangle} \langle x_i, x_j \rangle \in R^I$
$I \models G_1 \lhd G_2$ iff
$\forall x_1, \ldots, x_{\ell_2} \in \triangle^I : \langle x_1, \ldots, x_{\ell_1}, \ldots, x_{\ell_2} \rangle \in G_2^I \rightarrow \langle x_1, \ldots, x_{\ell_1} \rangle \in G_1^I$
$I \models G_1[v_1, \ldots, v_n] \leftrightarrow G_2[w_1, \ldots w_n]$ iff, for each $1 \leq i \leq n$,
$\forall x_1, \ldots, x_{\ell_1}, y_1, \ldots, x_{\ell_2} \in \triangle^I$:
$\langle x_1, \ldots, x_{\ell_1} \rangle \in G_1^I \wedge \langle y_1, \ldots, y_{\ell_2} \rangle \in G_2^I \wedge x_{v_i} = y_{w_i} \rightarrow \bigwedge_{1 \leq j \leq n} x_{v_j} = y_{w_j}$
Note: $\ell_{(i)}$ is the arity of the description graph $G_{(i)}$.

The main reasoning problem is satisfiability checking, as subsumption and instance checking can be reduced to satisfiability as usual.

4 Other Applications

Our formalism is applicable not only to anatomy, but to all domains in which the number of arbitrarily interconnected objects has a natural bound. In this section, we provide a few additional examples of domains that cannot be faithfully represented using OWL, but which could be modeled using our formalism.

Chemistry. The precise description of molecules is an important problem in bioinformatics [6]. A formal representation of molecules and chemical compounds is often used to integrate information from different chemical databases [6]. The structure of molecules is often not tree-like. For example, hydrocarbons are chemical compounds containing (often tree-like) carbon–hydrogen chains. Benzene is a hydrocarbon whose molecules contain at least one benzene ring (see Figure 3), and its structure can be described using our formalism: description graphs can be used to represent the benzene ring (which is bounded in size), while standard OWL axioms can be used to represent tree-like carbon–hydrogen chains.

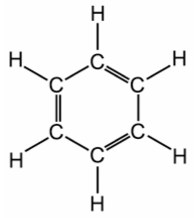


Fig. 3: The Benzene Ring

Scientific Workflows. Scientific workflows are descriptions of steps of scientific experiments, and they are often represented as directed graphs in which each node depicts a single experiment step and each edge represents information flow between two steps. The precise description of workflows is increasingly important, for example, in bioinformatics. There have been attempts to provide semantics to workflows using OWL [7], but their success has been rather limited so far due to their non-tree-like structure. Since workflows are typically bounded, however, they can naturally be represented using description graphs.

Engineering. OWL has recently been used in engineering domains, such as the aerospace industry, which involve the representation of very complex structured objects, such as aircrafts. The number of parts needed to describe an aircraft is naturally bounded (in the same way as it is in the case of a human), so such domains can easily be represented using description graphs.

5 Technical Results

We now summarize the main results about reasoning with graph-extended KBs. The first relevant result is the undecidability of the satisfiability problem: the interaction between DL axioms and rules alone, or between graphs and rules, or between DL axioms and graphs already leads to undecidability.

Theorem 1. *Checking satisfiability of graph-extended KBs* $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ *is already undecidable in the following situations:*

- $\mathcal{K} = (\mathcal{T}, \emptyset, \mathcal{G}, \mathcal{A})$ *with* $\mathcal{T}$ *a TBox in* $\mathcal{ALCF}$ *and* $\mathcal{G} = (\mathcal{G}_G, \emptyset, \emptyset)$*;*
- $\mathcal{K} = (\emptyset, \mathcal{P}, \mathcal{G}, \mathcal{A})$ *with* $\mathcal{P}$ *a Horn program and* $\mathcal{G} = (\mathcal{G}_G, \emptyset, \emptyset)$*;*
- $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \emptyset, \mathcal{A})$ *with* $\mathcal{P}$ *a Horn program and* $\mathcal{T}$ *in* $\mathcal{ALC}$*.*

Undecidability is partly due to the fact that a GBox can easily axiomatize existence of unbounded chains of description graph instances. As explained in the introduction, however, many domains can be described by arranging the description graphs in an acyclic hierarchy. We can reflect this hierarchy in a GBox by imposing on it the following acyclicity condition.

Definition 7 (Acyclic GBox). *A GBox* $\mathcal{G} = (\mathcal{G}_G, \mathcal{G}_S, \mathcal{G}_A)$ *is* acyclic *if a strict order* $\prec$ *on* $\mathcal{G}_G$ *exists s.t., for each* $G = (V, E, \lambda, M)$ *and* $G' = (V', E', \lambda', M')$ *in*

$\mathcal{G}_G$, *if* $G \not\preceq G'$, *then, for each* $A \in M'$ *and* $\lhd$ *the reflexive–transitive closure of* $\lhd$ *in* $\mathcal{G}_S$*:* (i) *if* $G' \lhd G$, *then* $\neg A \in \lambda\langle i \rangle$ *for each* $i \in V \setminus V'$*;* (ii) *if* $G' \not\lhd G$, *then* $\neg A \in \lambda\langle i \rangle$ *for each* $i \in V$*.*

A graph-extended knowledge base is *acyclic* if its GBox is acyclic. Intuitively, $G_1 \prec G_2$ means that G_2 is subordinate to G_1. In our example, we would have $G_{hand} \prec G_{finger}$ and $G_{hand} \prec G_{thumb}$, since the structures of the finger and the thumb are subordinate to the structure of a hand, respectively. We would also have $G_{finger} \prec G_{thumb}$, since a finger is more general than the thumb. The conditions in Definition 7 ensure that the existence of an instance of G_{hand} can imply existence of an instance of G_{finger}, but not vice versa. More generally, a description graph G_1 can imply existence of G_2 only if $G_1 \prec G_2$, which is compatible with the intuition of hierarchic decomposition of the domain.

Unfortunately, acyclicity is not sufficient to regain decidability To this end, we have proposed to place restrictions on the usage of atomic roles in $\mathcal{T}$, $\mathcal{P}$ and $\mathcal{G}$ in order to limit the possible interaction between different types of axioms.

Definition 8 (Role Separation). *A* role separation scheme Λ *is a triple of the form* $(N_\mathcal{T}, N_\mathcal{P}, N_\mathcal{G})$ *where* $N_\mathcal{T}$*,* $N_\mathcal{P}$*, and* $N_\mathcal{G}$ *are (not necessarily disjoint) sets of atomic roles. The roles in* $N_\mathcal{T}$*,* $N_\mathcal{P}$*, and* $N_\mathcal{G}$ *are called* $\mathcal{T}$*-,* $\mathcal{P}$*-, and* $\mathcal{G}$*-roles, respectively. A KB* $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ *is* Λ-separated *if all roles occurring in* $\mathcal{T}$*,* $\mathcal{P}$*, and* $\mathcal{G}$ *are* $\mathcal{T}$*-,* $\mathcal{P}$*-, and* $\mathcal{G}$*-roles, respectively. We say that* $\Lambda = (N_\mathcal{T}, N_\mathcal{P}, N_\mathcal{G})$ *is* weak *if* $N_\mathcal{T} \cap N_\mathcal{P} = \emptyset$*; it is* strong *if additionally* $N_\mathcal{G} = N_\mathcal{P}$*. A knowledge base* $\mathcal{K}$ *is* weakly separated *(respectively* strongly separated*) if a weak (respectively strong) role separation scheme* Λ *exists such that* $\mathcal{K}$ *is* Λ*-separated.*

Intuitively, weak separation prevents any interaction between $\mathcal{T}$ and $\mathcal{P}$. It allows one to describe general knowledge using TBox axioms and then to specialize such knowledge using graphs. For example, even if the general structure of a finger were described using DLs (e.g., this description might be a part of a general, coarse-grained KB that does not use graphs), one could describe more specialized knowledge, such as the structure of an index finger, using graphs. One can thus choose the appropriate style of modeling for knowledge at different levels of granularity. The main restriction is that one cannot use rules involving roles occurring in DL axioms. Acyclicity and weak separation seem reasonable assumptions in all the application domains mentioned in Section 4.

Strong separation restricts the modeling style in a more significant way than weak separation: essentially, it requires the modeler to determine in advance which knowledge will be modeled using DLs and which using graphs. Thus, knowledge modeled using DLs cannot be specialized using graphs and vice versa. The restriction to strong separation is particularly limiting in the use case of chemical compounds in Section 4. It is, however, reasonable in the anatomy and engineering use cases.

Theorem 2. *Checking satisfiability of an acyclic graph-extended knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{P}, \mathcal{G}, \mathcal{A})$ *is*

- *undecidable if* $\mathcal{K}$ *is weakly separated and* $\mathcal{T}$ *is in* $\mathcal{ALCIF}$*,*

– *decidable if* $\mathcal{K}$ *is strongly separated and* $\mathcal{T}$ *is in* $\mathcal{SHOIQ}$*,*
– NExpTime*-complete if* $\mathcal{K}$ *is weakly separated and* $\mathcal{T}$ *is in* $\mathcal{SHOQ}$*, or if* $\mathcal{K}$ *is strongly separated and* $\mathcal{T}$ *is in* $\mathcal{SHOQ}$ *or* $\mathcal{SHIQ}$*.*

In the case of strongly separated $\mathcal{K}$ and $\mathcal{T}$ in $\mathcal{ALCIF}$, the interaction between the inverse and functional roles leads to undecidability; in contrast, decidability is much more robust in the case of strong separation.

In [5, 2] we have presented several practical reasoning algorithms for the decidable cases identified in Theorem 2. Furthermore, we have implemented the formalism in the HermiT[4] reasoner [8]. Our preliminary performance evaluation has shown that our reasoning algorithms can solve many nontrivial problems.

6 Future Work

The main challenge for our future work is to validate the applicability of our formalism in applications. To this end, we will extend Protégé 4 to support description graphs and apply our formalism in the practical scenarios. We will also improve the implementation of our reasoning algorithms.

References

1. Vardi, M.Y.: Why Is Modal Logic So Robustly Decidable? In: Proc. of a DIMACS Workshop on Descriptive Complexity and Finite Models. (1996)
2. Motik, B., Grau, B.C., Sattler, U.: Structured Objects in OWL: Representation and Reasoning. In: Proc. of WWW 2008, ACM Press (2008) 555–564
3. Horrocks, I., Patel-Schneider, P.F.: A Proposal for an OWL Rules Language. In: Proc. of the 13th Int. World Wide Web Conference (WWW 2004), New York, NY, USA, ACM Press (May 17–22 2004) 723–731
4. Levy, A.Y., Rousset, M.C.: Combining Horn Rules and Description Logics in CARIN. Artificial Intelligence **104**(1–2) (1998) 165–209
5. Motik, B., Grau, B.C., Horrocks, I., Sattler, U.: Representing Structured Objects using Description Graphs. In: Proc. of KR 2008, AAAI Press (2008)
6. Konyk, M., Battista, A.D.L., Dumontier, M.: Chemical knowledge for the semantic web. In: DILS. Volume 5109 of LNCS., Springer (2008) 169–176
7. Goderis, A., Sattler, U., Goble, C.A.: Applying description logics for workflow reuse and repurposing. In: Proc. of DL. (2005)
8. Motik, B., Shearer, R., Horrocks, I.: Optimized Reasoning in Description Logics using Hypertableaux. In: Proc. CADE-21. (2007) 67–83 HermiT website: http://web.comlab.ox.ac.uk/oucl/work/boris.motik/HermiT.

[4] `http://web.comlab.ox.ac.uk/oucl/work/boris.motik/HermiT/`