Orchestrating a Network of Mereo(topo)logical Theories

C. Maria Keet
Department of Computer Science
University of Cape Town
Cape Town, South Africa
mkeet@cs.uct.ac.za

Oliver Kutz
KRDB Research Centre
Free University of Bozen-Bolzano
Bolzano, Italy
Oliver.Kutz@unibz.it

ABSTRACT

Parthood is used widely in ontologies across subject domains. Some modelling guidance can be gleaned from Ontology, yet it offers multiple mereological theories, and even more when combined with topology, i.e., mereotopology. To complicate the landscape, decidable languages put restrictions on the language features, so that only fragments of the mereo(topo)logical theories can be represented, yet during modelling, those full features may be needed to check correctness. We address these issues by specifying a structured network of theories formulated in multiple logics that are glued together by the various linking constructs of the Distributed Ontology Language, DOL. For the KGEMT mereotopological theory and five sub-theories, together with the DL-based OWL species and first- and second-order logic, this network in DOL orchestrates 28 ontologies. Further, we propose automated steps toward resolution of language feature conflicts when combining modules, availing of the new 'OWL classifier' tool that pinpoints profile violations.

CCS CONCEPTS

• **Computing methodologies** → **Knowledge representation and reasoning**; **Ontology engineering**; *Description logics*;

KEYWORDS

Mereology, Mereotopology, OWL, Ontology Engineering, Distributed Ontology Language DOL

ACM Reference format:
C. Maria Keet and Oliver Kutz. 2017. Orchestrating a Network of Mereo(topo)-logical Theories. In *Proceedings of K-CAP 2017: Knowledge Capture Conference, Austin, TX, USA, December 4–6, 2017 (K-CAP 2017),* 9 pages.
https://doi.org/10.1145/3148011.3148013

1 INTRODUCTION

Mereology, the theory of parthood, is well-established in Ontology (philosophy), and is used widely in applied ontology and ontologies on the Semantic Web and other ontology-driven information systems. For instance, the medical terminologies openGalen with its 23 part-whole relations [31] and SNOMED CT, the Gene Ontology [11] that is in widespread use for instance-level database integration in the biological sciences, and the Foundational Model of Anatomy [32] all use variants of mereology. Mereotopology is an extension of mereology with topological notions so as to be able to distinguish between interior part and tangential part. It is used in geographic information systems, annotation of pictures and with it, one can infer, e.g., whether a country is landlocked; e.g., [13, 21].

Due to the trade-offs between the expressiveness of logic languages and computational complexity, it has been difficult to represent mereology and mereotopology in full and such that one can obtain the desired inferences. Attempts include the SEP triples workaround [34], the extension of OWL with reflexivity and irreflexivity in $\mathcal{SROIQ}$ [17], and trade-off assessments for the OWL species on consequences for automated reasoning for mereotopology [21]. The most expressive Description Logics-based OWL language, OWL 2 DL, creates further complications for the modeller due to expressiveness limitations on object properties as a consequence of computational complexity trade-offs. A concrete example of such a trade-off is the choice between parthood's transitivity or its use with qualified number restrictions. This gives a modeller three options, using humans with their limbs and feet as example: 1) humans can have as part any number of limbs and infer that if a foot is part of a limb and a limb part of a human, then that foot is part of that human; 2) a (canonical) human has as part exactly four limbs but it cannot be inferred that the foot is part of the human; and 3) a human has exactly four limbs and we can make the (transitive) inferences about feet at the cost of scant tool support and poor performance compared to options 1 and 2, due to translating it to first-order logic (FOL) and calling a corresponding reasoner. If some OWL ontology O_1 uses option 1 and OWL ontology O_2 option 2, then importing or merging the ontologies leads to option 3—an OWL file outside of OWL 2 DL—and undecidability in general. This lack of closure under modular combination is a rather unusual aspect for a logic-based modelling language (notice that this problem does not exist in, e.g., FOL), and in the typical situation of ontology development with Protégé, this leaves the modeller stranded.

In sum, within the OWL context, it is confusing modelling as to which mereo(topo)logical theory to include, it does not meet the representation and reasoning requirements of the domain experts, computationally incompatible modelling choices may not be obvious, and yet for various scenarios different choices are applicable.

We aim to solve these issues by tying together two components. First, we will structure the mereological theories, using a two-pronged approach. There are several recognised sub-theories in the KGEMT mereotopology and there are established languages with their language features that can represent various subsets of those theories. We investigate this intersection and elucidate the maximum possible sub-theories for each language. They and their

K-CAP 2017, December 4–6, 2017, Austin, TX, USA

ACM ISBN 978-1-4503-5553-7/17/12…$15.00
https://doi.org/10.1145/3148011.3148013

interactions are presented formally in a networked set of theories represented in the Distributed Ontology Language (DOL), as it was specifically designed for linking theories represented in different logics (up to higher order logic). DOL is supported by a tool ecosystem (ontohub.org and Hets, see below) to select the appropriate reasoners for it, rather than deterministically sticking to one upfront. This resulted in 28 interconnected micro-ontologies—serving as basic versions of ontology design patterns—of mereology, topology, and mereotopology. A modeller then can choose precisely which theory to include for which usage scenario of the ontology, and when to use the linking to a more expressive version when reasoning time is not crucial. Because of the interactions of language features that pushes an ontology into one fragment or another, we propose steps towards *conflict resolution*, which pinpoints, automatically, which axioms violate which logic, so that a modeller can make an informed decision.

In the remainder of the paper, we first summarise mereotopology (Section 2) and the logics used for formalisation as well as DOL as a structuring language (Section 3), and subsequently outline the DOL techniques for the network formalisation (Section 4). The contribution to conflict resolution is described in Section 5. We discuss in Section 6 and conclude in Section 7.

2 PRELIMINARIES: MEREOTOPOLOGY

This summary of mereotopology is based on [36] and [21], which focuses on going from the basic axioms making up the simplest theories of parthood (M) and location (T) up to the KGEMT mereotopological theory, as depicted in Fig. 1.

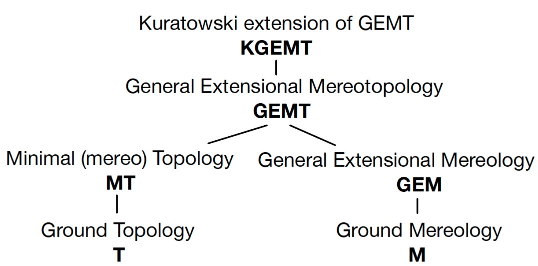

Figure 1: Hasse Diagram of the theories we focus on in this paper, from weaker (T, M) to stronger (after Varzi (2007)).

On the mereology side, the basic theory is Ground Mereology, M, that has part of as primitive, which is reflexive, antisymmetric, and transitive (t1, t2, t3 in Table 1). From this, one can define proper parthood (t20 in Table 2), from which irreflexivity (t25) can be deduced, and, following from that (antisymmetry of P + irreflexivity of PP), asymmetry (t27); proper part of is also transitive (t26). Overlap can now also be defined (t21). One can add the notion of supplementation (among other things) to M, resulting in General Extensional Mereology, GEM (M + t4, t5).

In the other branch, we begin with Ground Topology (T) with the connection relation, which is reflexive and symmetric (t6, t7). This can be extended to Minimal Topology (MT) by adding spatial enclosure (t8) as defined in t9. MT is then combined with GEM into GEMT, which consists of MT + GEM + t10 (converse monotonicity), t11 (self-connected), t12 (bridging connection to part), and t13 (fusion). With the GEMT axioms and definitions, one can then define interior proper part (t24), and from that, tangential proper part (t23). The final aspect is then about closure, interior, and exterior, resulting in KGEMT, i.e., GEMT + t14, t15, t16. The three extra axioms require their definitions (t17-t19), so they then also belong to KGEMT.

It is possible to construct a mereotopological theory in a different way, such as taking proper part of as primitive, or merging part of and connection into a ternary relation, or adding atomicity or boundaries (see [36] for details). There are also multiple options to incorporate more comprehensive topological theories; e.g., starting from connection to define other relations (including parthood) [30], focus on containment [4] or convex hulls [14] (see also Table 1 in [15] for possible ontological commitments). Our scope is not mereotopology *per sé*, however, but to find a usable way and a reusable approach to represent at least one such set of interconnected theories computationally and reason over it automatically.

Table 1: Axiomatization of KGEMT core axioms and definitions (based on [21], summarised from [36]). P: part of; **PP:** proper part of; **O:** overlap, **C:** connection; **E:** enclosure; **EQ:** indiscernible; **IPP:** interior proper part of; **TPP:** tangential proper part of; **SC: self-connected; c: closure; i: interior; e: exterior; +: sum; ~: complement.**

$P(x,x)$	(t1)
$P(x,y) \land P(y,z) \to P(x,z)$	(t2)
$P(x,y) \land P(y,x) \to x = y$	(t3)
$\neg P(y,x) \to \exists z(P(z,y) \land \neg O(z,x))$	(t4)
$\exists w\phi(w) \to \exists z \forall w(O(w,z) \leftrightarrow \exists v(\phi(v) \land O(w,v)))$	(t5)
$C(x,x)$	(t6)
$C(x,y) \to C(y,x)$	(t7)
$P(x,y) \to E(x,y)$	(t8)
$E(x,y) =_{df} \forall z(C(z,x) \to C(z,y))$	(t9)
$E(x,y) \to P(x,y)$	(t10)
$SC(x) \leftrightarrow \forall y,z(x = y + z \to C(y,z))$	(t11)
$\exists z(SC(z) \land O(z,x) \land O(z,y) \land \forall w(P(w,z) \to (O(w,x) \lor O(w,y)))) \to C(x,y)$	(t12)
$z = \sum x \phi x \to \forall y(C(y,z) \to \exists x(\phi x \land C(y,x)))$	(t13)
$P(x,cx)$	(t14)
$c(cx) = cx$	(t15)
$c(x+y) = cx + cy$	(t16)
$cx =_{df} \sim (ex)$	(t17)
$ex =_{df} i(\sim x)$	(t18)
$ix =_{df} \sum z \forall y(C(z,y) \to O(x,y))$	(t19)

Finally, note that such a KGEMT theory does not (have to) exist in isolation. KGEMT with its relations in [21] was extended from the basic taxonomy of part-whole relations of [20] that uses DOLCE [26] for domain and range restrictions. DOLCE incorporates GEM [26], so KGEMT would be compatible with DOLCE, effectively extending it. Other foundational ontologies rely heavily on theories of parts, notably GFO [16] and BFO[1], who are struggling to reconcile the expressive theories from Ontology with the practicalities of

[1] http://ifomis.uni-saarland.de/bfo/

Table 2: Basic additional axioms, definitions, and theorems (based on [21], summarised from [36]).

$PP(x,y) =_{df} P(x,y) \land \neg P(y,x)$	(t20)
$O(x,y) =_{df} \exists z(P(z,x) \land P(z,y))$	(t21)
$EQ(x,y) =_{df} P(x,y) \land P(y,x)$	(t22)
$TPP(x,y) =_{df} PP(x,y) \land \neg IPP(x,y)$	(t23)
$IPP(x,y) =_{df} PP(x,y) \land \forall z(C(z,x) \to O(z,y))$	(t24)
$\neg PP(x,x)$	(t25)
$PP(x,y) \land PP(y,z) \to PP(x,z)$	(t26)
$PP(x,y) \to \neg PP(y,x)$	(t27)

OWL and OBO; e.g., the first order logic version of BFO includes a universal parthood theory that includes a partonomic inclusion theory[2] yet this is absent from its OWL and OBO version. Finally, it could serve the many attempts in domain ontology development in subject domains such as medicine and environmental data [9, 13].

3 PRELIMINARIES: LOGICS

While many logics could be considered, we zoom into those with a considerable uptake and some tool support: the OWL family of languages [7, 18], FOL [19], and the newly standardised DOL metalanguage [24, 28].

3.1 Mereotopology and OWL

The latest standardised language in the OWL family is OWL 2 [29], of which most 'species' in the family are based on Description Logics (DLs) [2]. DLs are decidable fragments of FOL, aiming to find 'sweet spots' of the trade-offs between expressiveness and decidability; e.g., with respect to full FOL, it could prohibit the use of negation or the use of number restrictions greater than one so as to obtain a language that is computationally better behaved. In general, most DLs do not have extensive support for relational properties. In fact, it is already problematic to represent even just Ground Mereology fully, as shown in Table 3 for the DL-based OWL species. The "+"'s in the OWL 2 DL column are deceptive: a transitive property is not 'simple' anymore—simple properties are those that are not transitive or defined by property chains—so cannot be used jointly with properties that involve negation (i.e., irreflexivity and asymmetry) or appear in qualified cardinality restriction. This causes an increase of subtheories, as we shall see further below (e.g.: theory 15 vs. 16 in Table 4).

3.2 Relevant DOL features

In a situation where two independently developed ontologies are supposed to be reused as modules in a larger ontology, the differences between these ontologies will typically prevent them from working together properly. Solving the interoperability issues involves the identification of synonyms, homonyms, and the development of bridge axioms to link the ontologies appropriately [5, 10]. Addressing these challenges, there is a diversity of notions providing design patterns for, and interrelations among, ontologies. The Distributed Ontology, Model and Specification Language (DOL)

[2] http://www.acsu.buffalo.edu/~bittner3/Theories/BFO/

Table 3: Properties of parthood ($.^{P}$) and proper parthood ($.^{PP}$) in Ground Mereology, and connection ($.^{C}$) in Ground Topology and their inclusion in the OWL family and FOL.

Language ⇒	DL-based OWL species						FOL
Feature ⇓	DL	Lite	2DL	2QL	2RL	2EL	
Symmetry^{C}	+	+	+	+	+	−	+
Reflexivity^{P,C}	−	−	+	+	−	+	+
Antisymmetry^{P}	−	−	−	−	−	−	+
Transitivity^{P,PP}	+	+	+	−	+	+	+
Asymmetry^{PP}	−	−	+	+	+	−	+
Irreflexivity^{PP}	−	−	+	+	+	−	+

aims at providing a unified metalanguage for handling this diversity. The general theoretical background for DOL is presented in [24] and a detailed description of the language can be found in [28]. It was approved as a standard of the Object Management Group (OMG) in 2016[3]. DOL enjoys the following distinctive features: (1) structuring constructs for building ontologies from existing ontologies, like imports, union, forgetting, interpolation, filtering, and open-world versus closed-world semantics; (2) module extraction; (3) mappings between ontologies, like interpretation of theories, conservative/definitorial extensions, alignments, etc.; and (4) networks of ontologies, and their combination.

DOL and its structuring language are designed as a multi-logic meta-language, already supporting all of the mainstream ontology languages in use today, as depicted in Fig. 2. The graph is organised along two dimensions: 1) 'quality of logic translation' and 2) expressivity of the logic. The expressivity ranges from RDF to OWL to variants of first- and second-order logic, exhaustively covering the modelling requirements of diverse communities. The framework is based on the theory of 'institutions', abstracting from the peculiarities of syntax and semantics of particular logics, see [12, 24] for technical details. The 'quality of translation' is related to how directly proof support can be guaranteed along the translation; e.g., a 'subinstitution' corresponds intuitively directly to a sub-logic of the target and a 'theoroidal subinstitution' requires one to encode some of the semantics of the source into extra axioms expressed in the target logic, for instance when encoding many-sorted logic back into single sorted logic.

4 TYING IT TOGETHER WITH DOL

Having outlined the mereo(topo)logical theories and the various logics, then in combining the two, one can construct 28 theories, which are listed in Table 4. The 28 came about by, one by one, taking a theory listed in Fig. 1 with the axioms that make it up (see Sect. 2) and assess which of those can be represented in each of the selected languages. Note that there are not 2^{27} theories, because not every combination of the 27 axioms in Tables 1 and 2 makes sense ontologically: e.g., one would not want a theory consisting of, say, t1 (reflexivity of parthood) and t16 (exterior), because that combination is not meaningful. The converse—just the six named theories in Fig. 1—does not apply either, because even ground mereology cannot be represented fully in OWL 2 DL (recall Table 3); hence, it is also

[3] http://www.omg.org/spec/DOL/

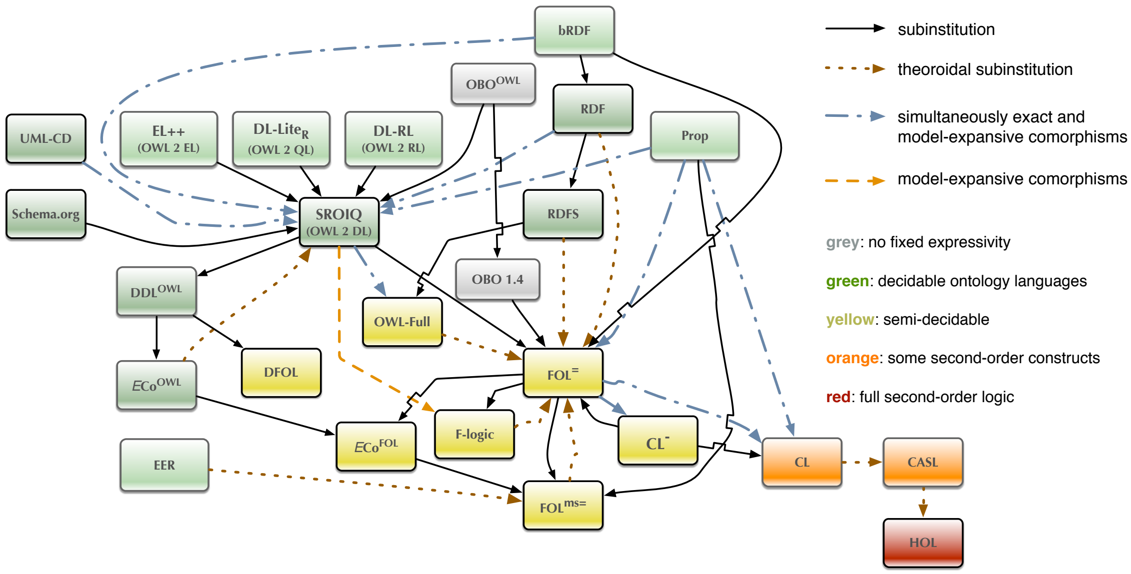

Figure 2: A sub-graph of logics currently supported by DOL/Ontohub, linked with a variety of logic translations.

not simply a "6 (mereo(topo)logical theories) × (6 (DL-based OWL species)+FOL+HOL) = 48" due to overlaps and exclusions in features among the languages, as discussed in Section 3.1. Finally, one cannot assert property definitions like t20 in OWL, whereas one can in FOL and HOL, and therefore they were added as primitives to the OWL-formalised theories. Theories 1–19 are already available on OntoHub at https://ontohub.org/repositories/mereotopology, and the repository is currently being extended with first- and second-order modelling and full structuring.[4]

For presentation purposes and anticipated usage, we present the structuring of the network of theories using DOL by focusing on the OWL species and extensions to OWL + FOL. From the DOL structuring point of view, second-order axioms can be dealt with in just the same way, including SOL (a second-order sublogic of CASL), HasCASL (the higher-order extension of CASL), Common Logic, Isabelle/HOL and HOL-light, as well as THF, see e.g., [27].

4.1 Organising micro-ontologies in DOL

The basic structuring operations for logical theories are already available on the logic-specific level. For instance, when working purely in OWL, we can employ the purely homogeneous DL-based OWL fragment of DOL, called DOWL [25]. The most basic mechanisms relevant for the present paper that we gain on top of OWL are the following:

(1) control over signatures via renaming symbols along imports,
(2) extending existing theories with new axioms (theory extensions),
(3) unions of theories,
(4) theory interpretation,
(5) syntactic extraction of modules using specified symbols, and
(6) lemma book-keeping and counterexample specification.

Features 1-5 will be illustrated in this section; the two more advanced DOL features (item 6) will be discussed in Section 4.2. We begin by illustrating the idea of extending a theory by new axioms (feature 2). We use the simplest possible examples on purpose, to illustrate the underlying ideas as clearly as possible.

Extensions. We extend theory8 (OWL 2 EL/QL) into theory4 (OWL 2 QL) by adding symmetry (t7). This is written as follows:

```
logic OWL2.QL
ontology theory4 =
theory8
then
ObjectProperty: C Characteristics: Symmetric %(t7)
```

We now simultaneously illustrate unions of theories (feature 4) with control over signatures (feature 1):

Unions. The union of theory7 (OWL 2 EL) and theory11 (OWL 2 RL) results in theory2, expressed in OWL 2 QL. To simultaneously illustrate how we can directly manage signatures, the given theories use diverging vocabulary for talking about connection, which is a very common phenomenon in practice.

```
logic OWL2.QL
ontology theory2 =
theory7 with Con |- > C
 and
theory11 with Co |- > C
```

The result is a theory that combines the axioms of theories 7 and 11, with a unified signature for connection, using the symbol *C*.

Note that regarding unions of OWL theories more generally, in the DOL framework, an OWL ontology that breaks the syntactic restrictions imposed by OWL 2 DL can still be equipped with sound and complete reasoning by translating it to FOL and calling a corresponding reasoner, see also [33].

[4] An ontology repository for FOL theories is COLORE, see http://stl.mie.utoronto.ca/colore/. It includes RCC and Asher and Vieu's system [1] and mereology, and relates them via theory extensions and interpretations, however not using explicit DOL structuring for this and not covering the OWL-based systems.

Table 4: Subsets of KGEMT that can be represented in HOL, FOL, and the OWL species. For the OWL species, t9, t20, t21, t22, t23, t24 were simplified and added as primitives (axiom number of Tables 1 and 2 are appended with a "p"). For readability, FOL and HOL are not listed where OWL species are listed, and OWL 2 DL is not listed if it lists an OWL 2 DL fragment.

N	Language	Subsets of KGEMT axioms	Comments
1	OWL 2 QL	t1, t21p, t22p	M, with p, partially
2	OWL 2 QL	t6, t7	T, c
3	OWL 2 QL	t20p, t21p, t22p, t25, t27	M, pp
4	OWL 2 QL	t6, t7, t8, t9p	MT
5	OWL 2 QL	t1, t6, t7, t8, t9p , t10, t20p, t21p, t22p, t23p, t24p, t25, t27	GEMT, partial
6	OWL 2 EL, 2 QL	t1, t2, t21p, t22p	M, with p, partially
7	OWL 2 EL	t6	T, c, partial
8	OWL 2 EL, 2QL	t6, t8, t9p	MT, partially
9	OWL 2 EL	t1, t2, t6, t8, t9p, t10, t26, t20p, t21p, t22p, t23p, t24p	GEMT, partial
10	OWL 2 RL, OWL Lite, DL	t2, t21p, t22p	M, p, partial
11	OWL 2 RL, 2QL, OWL Lite, DL	t7	T c, partial
12	OWL 2 RL, EL, DL, OWL Lite, DL	t2, t26, t20p, t21p, t22p	M, with p and pp both partially
13	OWL 2 RL, OWL Lite, DL	t7,t8, t9p	MT partial
14	OWL 2 RL, OWL Lite, DL	t2, t7, t8, t9p, t10, t26, t20p, t21p, t22p, t23p, t24p	GEMT, partial
15	OWL 2 DL	t1, t2, t6, t7, t8, t9p, t10, t25, t27, t20p, t21p, t22p, t23p, t24p	GEMT, partial
16	OWL 2 DL	t1, t2, t6, t7, t8, t9p, t10, t26, t20p, t21p, t22p, t23p, t24p	GEMT, partial
17	OWL 2 RL	t2, t20p, t21p, t22p, t25, t27	M with p and pp, both partial
18	OWL 2 DL	t1, t2, t25, t27, t20p, t21p, t22p	M with p and pp, both partial
19	OWL 2 EL	t1, t2, t26, t20p, t21p, t22p	M with p and pp, partial
20	FOL, HOL	t1, t2, t3, t21, t22, t4	M, with p
21	FOL, HOL	t1, t2, t3, t20, t21, t22, t25, t26, t27	M, with p and pp
22	FOL	t1-t4, t20, t21, t22, t25, t26, t27	GEM, partial
23	FOL, HOL	t6, t7, t8, t9	MT
24	FOL	t1-t4, t6-t12, t20-t27	GEMT, partial
25	FOL	t1-t4, t6-t12, t14-t27	KGEMT, partial
26	HOL	t1-t5, t20, t21, t22, t25, t26, t27	GEM
27	HOL	t1-t13, t20-t27	GEMT
28	HOL	t1-t27	KGEMT

Theory Interpretation. To continue the previous example, clearly, theory7 (or theory11) can be interpreted into theory2 in the sense that theory2 can prove all the consequences of theory7. This is written as follows:

```
interpretation theory7_into_theory2 :
        theory7 to theory2 =
        Con ↦ C
```

The semantics is that under the translation of taking *Con* to *C*, theory2 can prove all the consequences of theory7.

Heterogeneous DOL*: Logic translation.* The heterogeneous case, i.e., when moving from a weaker logic (say, OWL DL) to a more expressive one (say, full FOL) is a specific strength of DOL. We illustrate this by extending theory6 (in OWL 2 EL) with antisymmetry and weak supplementation (this is 'almost' theory20 (in FOL), however still lacking the definitions of *O* and *EQ*), as follows:

```
logic CASL.FOL
ontology theory6_plus_antisym_and_WS =
theory6 with translation OWL22CASL
then
forall x,y:Thing . P(x,y) /\ P(y,x) => x =y %%(t3)
forall x,y:Thing . not P(y,x) =>
      exists z:Thing . P(z,y) /\ not O(z,x) %%(t4)
```

Definitional Extensions. Definitional extensions are one of the most basic tools in ontology design (cf., e.g., DOLCE) and can be specified explicitly in DOL. Indeed, they are one of the basic structuring means to organise mereotopological theories: e.g., proper part of (PP) and overlap (O) can be defined in terms of part of (P), and tangential proper part of (TPP) in terms of part of and connection (see Table 2).

Technically, a definitional extension with a definition of, e.g., overlap (O) consists of a signature extension with the binary predicate symbol $O(x, y)$, together with the basic definition of O in terms of P. More formally, a theory T_2 in signature σ' is a *definitional extension* of theory T_1 in signature σ, if any T_1-model has a unique expansion to a T_2-model.[5] Intuitively, T_2 adds neither additional constraints nor additional freedom of interpretation to T_1, but rather the new symbols in T_2 are uniquely defined in terms of the symbols in T_1 [24]. Note that it is a slightly different situation when previously introduced symbols, even if unaxiomatised, are *augmented* with definitions (which is in general neither definitional nor conservative). We say that T_2 is a *weakly definitional theory extension* of T_1

[5]Note that this means that model reduct is a bijection between T_1-models and T_2-models.

if each realisation of T_1 can be expanded to *at most one* realisation of T_2. All these requirements can be expressed in DOL (definitional extensions are annotated with `%def`, weak definitional extensions with `%wdef`). We complete the previous incomplete definition of theory20 as follows:

```
logic CASL.FOL
ontology theory20 =
theory6_plus_antisym_and_WS
 then %wdef
. forall x,y:Thing . O(x,y) <=> exists z:Thing  (P(z,x)
                                         /\ P(z,y)) %(t21)
. forall x,y:Thing . EQ(x,y) <=> P(x,y) /\ P(y,x) %(t22)
```

The result is a complete specification of theory20, where the previously undefined symbols O and EQ have been augmented with their formal definitions.

DOL *network and colimits.* The entire network of 28 theories can be formally represented as a DOL network, written as follows:

```
network KGEMT_network = theory1,...,theory28,M1,...,Mp
```

Here, the theories theory1–theory28 are listed, followed by the defined theory interpretations Mi connecting the various ontology nodes.

Since the network explores the subtheories of KGEMT along the dimensions of language expressivity and ontological soundness, there are no conflicting logical statement among the 27 axioms. In fact, DOL's combination technique (based on computing colimits of the network) can compute the colimit of the network of theory1–theory28 resulting in the full KGEMT.

```
ontology KGEMT = combine KGEMT_network
```

Moreover, the network construct allows us to exclude certain mappings and ontology nodes. Therefore, we can select a language species, and compute the maximal combination of subtheories that live in that species. Note, as discussed earlier, that this might lead outside of the logic species for some logics, such as OWL 2 DL, but this will not occur for OWL 2 EL or FOL sub-networks.

4.2 Lemmas, Consistency, and Countermodels

To understand the essential features of our network of 28 mereotopological theories, it is important to (a) keep track of the desired consequences of the theories, and (b) to record counterexamples of properties that do *not* follow from the current theory. Both features are supported by DOL, as discussed next.

Keeping track of consequences. Keeping track of desired consequences is easy in DOL. We continue the example from above. If proper part of is introduced definitorially, rather than axiomatised directly, then we expect its typical properties to be entailed by the definition. This, of course, depends on how precise we axiomatised the defining property of part of. We can augment any given theory with a definition of PP, and record the consequences, as follows:

```
logic CASL.FOL
ontology theory20_with_PP_lemmas =
theory20
 then %wdef
. forall x,y: Thing . PP(x,y) <=> P(x,y) /\ not P(y,x)
then  %implies
.forall x,y: Thing . not PP(x,x) %(t25)
.forall x,y,z: Thing . PP(x,y) /\ P(y,z) => PP(x,z) %(t26)
.forall x,y,z: Thing . PP(x,y) => not PP(y,z) %(t27)
```

The semantics is that, assuming the stated theory, the declared lemmas (sentences in the present logic), namely, the three axioms listed after the annotation `%implies` are logical consequences of the theory. This ontology specification therefore introduces so-called *proof obligations*, i.e., forces a connected automated reasoner (in this case SPASS or Vampire, etc.) to prove the axioms (t25)–(t27) from the theory specified before.

Relative consistency and countermodels. Proving consistency of the theories is easy for the OWL-based ones, for one simply can use one of the DL-based OWL 2 automated reasoners. Theories 1-19 have been checked as such, and are consistent according to those reasoners.

However, as soon as we move away from OWL to FOL, establishing consistency can be extremely difficult or in fact hopeless with current automatic reasoning capabilities. As reported in [23], even establishing the consistency of classical extensional parthood breaks (then) current automatic reasoning support unless model finders are strongly assisted by humans, e.g., as an oracle to prescribe the exact size of a model to look for.

We here outline how we can, within DOL, specify declaratively, and verify automatically, the consistency of adding an (or several) axiom to an existing theory, and related, how to prove that a theory admits unintended models, i.e., counterexamples for specific principles. The core tools to do this are a) theory interpretation, and b) formal specification of models. A model can be declaratively specified as a logical theory with exactly one model up to isomorphism. This is easy in propositional logic (fix the truth value of all propositions); in OWL we can specify partial models with the help of ABoxes; in FOL there are standard methods to describe finite models; and CASL can describe also infinite models via its free sort generation feature. For instance, to prove that theory23 is consistent, one specifies a finite model in a specification M which we assume uses symbols co, e, P and write the following interpretation:

```
logic CASL.FOL
interpretation Cons : theory23 to M = C |-> co, E |-> e
```

Notice that only the non-identical symbols need to be mapped explicitly. By definition of theory interpretation, 'Cons' is correct if and only if every model of M reduced along the morphism defined in Cons is a model of theory23. Now M specifies a fixed model and thus proving the interpretaton correct (the axioms of theory23 hold on M) establishes consistency.[6]

Notice that the same technique can be used to show that a theory T is both consistent and admits counterexamples to specific principles, by interpreting T into a model M that violates principle ϕ. Note that the heterogeneous features of DOL here become a powerful tool, since the countermodel can be specified in a more expressive logic (e.g., FOL) than the theory that we interpret (e.g., OWL EL). A very simple example: An OWL mereology leaving out antisymmetry (t3), say theory15, clearly has unintended models. We can specify in FOL a non-well-founded model for part of, and

[6]The solution to the problem of establishing consistency of 'too large' first-order theories proposed in [23] is to break down the global problem into a series of relative consistency proofs using the idea of CASL architectural specification [3] which prescribes how 'global models' can be put together from 'smaller models'. Then the small steps in this global consistency proof consist of constructions as the one just described.

by interpreting theory15 into this model, using the above technique, give formal proof that theory15 admits such models (but see [6] for philosophical arguments why one would like to drop antisymmetry).

5 TOWARD RESOLVING CONFLICTS BETWEEN LANGUAGES AND THEORIES

Trade-offs between representational expressivity, and therewith the possible inferences and reasoning efficiency, are shown clearly in Table 4, ranging from the OWL 2 profiles that have complexity for the satisfiability problem in polynomial time to the highly undecidable HOL. When a particular fragment is chosen for an ontology, it can be sent to the appropriate automated reasoners, such as CEL or ELK for OWL 2 EL, HOL reasoners such as Leo, Satallax, Isabelle (interfaced through Hets), and so on. This is an engineering task now that the options are known, and therefore not pursued here. It does move the goalpost to a form of conflict resolution to decide about what to do with *conflicting language features* of the languages used to represent the ontologies. For instance, one wants to methodically create an ontology network [35] by linking several ontologies, where OWL ontology O_1 uses theory18 and ontology O_2 uses theory19. However, combining them violates the OWL 2 DL RBox restrictions [29] and thus will lead to an undecidable language (FOL). This is a conflict if the requirement is to remain within OWL 2 DL expressiveness and a modeller has to know about it, why or what is causing it, and then decide how to resolve it, and likewise for an algorithm to determine to which optimised automated reasoner to send the ontology.

The main steps in the conflict resolution are: 1) (automatically) detecting language violations; 2) reporting which axioms use what feature that is beyond a particular language; 3) assist with describing the consequences of the options one has at disposal. Here, we focus on solving the first two steps, and scoping its feasibility in particular. To make the proof-of-concept manageable, an OWL species classifier was developed[7], which analyses the OWL file, lists *in which OWL species* it is and *why it violates the others*. The OWL classifier uses version 1.4.3 of the OWL API for OWL 1 in order to be able to categorise it in an OWL 1 species and OWL API v4.2.3 for all OWL 2 species, and it extends the DLExpressivityChecker class in the OWL API to keep track of axioms w.r.t. construct letters and to create a justification of the reported expressivity, therewith providing features not available in currently popular ontology editors. For instance, merging an ontology O with theory8 into O' that has theory13 results in an ontology in, at least, OWL 2 QL, 2 DL or Full (see Fig. 3), thus neither in the 2 EL or 2 RL profiles that the original ones were, respectively. The "Profile violations" section (see Fig. 3) can be used for resolving feature/computational complexity conflicts, showing which axiom(s) one would have to delete in order to stay within a particular profile, or, as in combining theory18 with theory19, which one(s) cause it to go beyond OWL 2 DL.

Regarding step 3 of the conflict resolution ('what to do'?), it is not possible to perform this task automatically, for it depends on what the modeller wants. At best, one can describe the options and their consequences. For the afore-mentioned example with theory no. 18 and 19, that would be:

(1) Choosing O_1's axioms with irreflexivity and asymmetry on proper part of will make the ontology interoperable with other ontologies represented in OWL 2 DL, FOL or HOL.
(2) Choosing O_2's axioms, with transitivity on part of and proper part of, will facilitate linking to ontologies in OWL 2 RL, 2 EL, 2 DL, FOL, and HOL.
(3) Choosing to keep, and combine, both sets will result in an OWL 2 Full ontology that is undecidable, and is compatible with FOL and HOL ontologies.

Such options with their consequences can be distilled from the set of KGEMT subtheories for the languages considered, and their formal specifications. Computing these options and from those results generating some natural language sentences may be in the realm of the feasible. However, a complicating factor is that this does not yet consider the other knowledge represented in the ontology that may interfere with this seemingly clean scenario, but would have to be taken into account. This is left for future work.

6 DISCUSSION

The many identifiable theories with just a few logics suffices to demonstrate both the (perhaps untenable) 'blow up' of mini-theories and why it has become confusing for modellers, yet also a way for catering for it within DOL. It is, of course, possible to further extend the set of mini-theories by considering also other DLs and other intermediate mereological theories (e.g., Extensional Mereology, EM), extending it with boundaries, and so on. It thus also provides the mechanism to verify computationally further extensions to KGEMT, such as KGEMTS [14]. While it may not be trivial at this stage to represent all this in DOL, the previous section demonstrates that it is feasible to do. The chance that modellers of any two ontologies have made the same choice as to which subset of axioms to include is small, and thus importing one into the other easily will lead to going beyond the target language, as illustrated. This overview presented here clearly demonstrates the options at one's disposal, and a way to deal with them within one framework.

This fine-grainedness and 'feature negotiation' is an improvement over plain `owl:import` that requires one to simply import a whole ontology and be stuck with the consequences. It also compares favourably with respect to a related framework for theory combination, ε-connections [22], that enjoyed an implementation with OWL as "link properties" [8], because there one can link vocabulary elements across ontologies only, whilst here we are looking at combining theories also on object property characteristics.

Although the OWL species classifier covers only all OWL species and not also FOL/HOL negotiation, we expect this is the most useful nonetheless. The OWL species classifier is a proof-of-concept tool assisting with conflict resolution, rather than end-user level, but it is the first tool that clearly organises the necessary information to be able to do so.

7 CONCLUSION

We have presented a network of 28 modular ontologies in DOL, relating various fragments of the KGEMT mereotopological theory and its five sub-theories, which was driven by the expressiveness

[7] https://github.com/muhummadPatel/OWL_Classifier

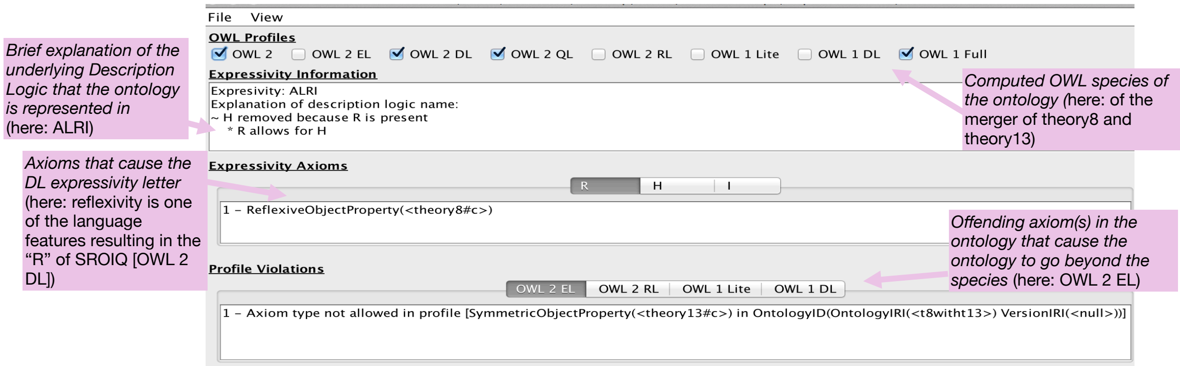

Figure 3: Annotated screenshot of the OWL species classifier output of the merger between theory8 and theory13.

of the six DL-based OWL species and first and second order logic. A core step toward resolution of feature conflicts among the OWL species was introduced with the OWL classifier, informing a modeller about language violations in order to make informed decisions. Future work includes extending this network with incompatible extensions, such as removing antisymmetry and adding atomicity, as well as a detailed specification of countermodels, formal consistency proofs, and the exploration of logic-specific sub-networks.

Acknowledgments The authors would like to thank Aashiq Parker, Brian Mc George, and Muhummad Patel for developing the OWL species classifier. We also thank Mihai Codescu and Fabian Neuhaus for discussing DOL features, and the anonymous reviewers for valuable feedback and corrections.

REFERENCES

[1] N Asher and L Vieu. 1995. Toward a Geometry of Common Sense: A Semantics and a Complete Axiomatization of Mereotopology. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal.*

[2] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider (Eds.). 2008. *The Description Logics Handbook* (2 ed.). Cambridge University Press.

[3] M. Bidoit, D. Sannella, and A. Tarlecki. 2002. Architectural Specifications in CASL. *Formal Aspects of Computing* 13 (2002), 252–273.

[4] T. Bittner and M. Donnelly. 2005. Computational ontologies of parthood, componenthood, and containment. In *Proc of IJCAI'05*. AAAI Press, 382–387.

[5] M. Codescu, T. Mossakowski, and O. Kutz. 2014. A Categorical Approach to Ontology Alignment. In *Proc. of OM'14*. CEUR-WS, Riva del Garda, Italy.

[6] A. J. Cotnoir. 2010. Anti-Symmetry and Non-Extensional Mereology. *The Philosophical Quarterly* 60, 239 (2010), 396–405.

[7] B. Cuenca Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler. 2008. OWL 2: The next step for OWL. *J Web Sem* 6, 4 (2008), 309–322.

[8] B. Cuenca Grau, B. Parsia, and E. Sirin. 2006. Combining OWL ontologies using ε-Connections. *J Web Sem* 4, 1 (2006), 40–59. https://doi.org/DOI:10.1016/j.websem.2005.09.010

[9] M. Donnelly, T. Bittner, and C. Rosse. 2006. A formal theory for spatial representation and reasoning in biomedical ontologies. *Artif Intell Med* 36, 1 (2006), 1–27.

[10] Jérôme Euzenat and Pavel Shvaiko. 2013. *Ontology Matching* (2nd ed.). Springer.

[11] Gene Ontology Consortium. 2000. Gene Ontology: tool for the unification of biology. *Nat Gen* 25 (2000), 25–29.

[12] J. A. Goguen and R. M. Burstall. 1992. Institutions: Abstract Model Theory for Specification and Programming. *J of the ACM* 39, 1 (1992), 95–146.

[13] R. Grütter and B. Bauer-Messmer. 2007. Combining OWL with RCC for Spatioterminological Reasoning on Environmental Data. In *Proc. of OWLED'07*. 6-7 June 2007, Innsbruck, Austria.

[14] M. Haemmerli and A. C. Varzi. 2014. Adding Convexity to Mereotopology. In *Proc. of FOIS'14*, Pawel Garbacz and Oliver Kutz (Eds.). IOS Press, 65–78.

[15] Torsten Hahmann and Michael Gruninger. 2012. Region-based theories of space: mereotopology and beyond. In *Qualitative Spatio-Temporal Representation and Reasoning: Trends and Future Directions*, Shyamanta M. Hazarika (Ed.). IGI-Global.

[16] H. Herre and B. Heller. 2006. Semantic foundations of medical information systems based on top-level ontologies. *Knowledge-Based Sys.* 19 (2006), 107–115.

[17] I. Horrocks, O. Kutz, and U. Sattler. 2006. The Even More Irresistible $\mathcal{SROIQ}$. *Proc. of KR'06* (2006), 452–457.

[18] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. 2003. From SHIQ and RDF to OWL: The making of a web ontology language. *J Web Sem* 1, 1 (2003), 7.

[19] I. Horrocks and A. Voronkov. 2006. Reasoning Support for Expressive Ontology Languages Using a Theorem Prover. In *Proc. of FoIKS'06 (LNCS)*. Springer, 201–218.

[20] C. M. Keet and A. Artale. 2008. Representing and Reasoning over a Taxonomy of Part-Whole Relations. *Appl Ont* 3, 1-2 (2008), 91–110.

[21] C. M. Keet, F. C. Fernández-Reyes, and A. Morales-González. 2012. Representing mereotopological relations in OWL ontologies with OntoPartS. In *Proc. of ESWC'12 (LNCS)*, E. Simperl et al. (Eds.), Vol. 7295. Springer, 240–254. 29-31 May 2012, Heraklion, Crete, Greece.

[22] O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. 2004. E-Connections of Abstract Description Systems. *Artif Intell* 156, 1 (2004), 1–73.

[23] O. Kutz and T. Mossakowski. 2011. A Modular Consistency Proof for Dolce. In *Proc. of AAAI'11*. AAAI Press, San Francisco, California.

[24] O. Kutz, T. Mossakowski, and D. Lücke. 2010. Carnap, Goguen, and the Hyperontologies: Logical Pluralism and Heterogeneous Structuring in Ontology Design. *Logica Universalis* 4, 2 (2010).

[25] O. Kutz, F. Neuhaus, M. M. Hedblom, T. Mossakowski, and M. Codescu. 2016. Ontology Patterns with DOWL: The Case of Blending. In *Proc. of DL'16*, Vol. 1577. CEUR-WS, Cape Town, South Africa, April 22-25, 2016.

[26] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. 2003. Ontology Library. WonderWeb Deliverable D18 (ver. 1.0, 31-12-2003).. (2003). http://wonderweb.semanticweb.org.

[27] T. Mossakowski, M. Codescu, O. Kutz, C. Lange, and M. Gruninger. 2014. Proof Support for Common Logic. In *Proc. of ARQNL'14*. July 23, Vienna Summer of Logic.

[28] T. Mossakowski, M. Codescu, F. Neuhaus, and O. Kutz. 2015. *The Road to Universal Logic–Festschrift for 50th birthday of Jean-Yves Beziau, Volume II.* Birkhäuser, Chapter The distributed ontology, modelling and specification language - DOL.

[29] B. Motik, P. F. Patel-Schneider, and B. Parsia. 2009. *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax.* W3C Recommendation. W3C. http://www.w3.org/TR/owl2-syntax/.

[30] D. A. Randell, Z. Cui, and A. G. Cohn. 1992. A spatial logic based on regions and connection. In *Proc. of KR'92*. Morgan Kaufmann, 165–176.

[31] J. Rogers and A. Rector. 2000. GALEN's model of parts and wholes: experience and comparisons. In *Proc. of AMIA Symp'00*. AMIA, 714–718.

[32] C. Rosse and J. L. V. Mejino Jr. 2003. A reference ontology for biomedical informatics: the foundational model of anatomy. *J Biomed Inf* 36, 6 (2003), 478–500.

[33] M. Schneider, S. Rudolph, and G. Sutcliffe. 2013. Modeling in OWL 2 without Restrictions. In *Proc of OWLED'13 (CEUR-WS)*, Vol. 1080.

[34] S. Schulz, U. Hahn, and M. Romacker. 2000. Modeling Anatomical Spatial Relations with Description Logics. In *Proc. of AMIA Symp'00*. 779–783.

[35] M. C. Suarez-Figueroa et al. 2008. *NeOn Methodology for Building Contextualized Ontology Networks.* NeOn Deliverable D5.4.1. NeOn Project.

[36] A.C. Varzi. 2007. *Handbook of Spatial Logics.* Berlin Heidelberg: Springer Verlag, Chapter Spatial reasoning and ontology: parts, wholes, and locations, 945–1038.