Three Semantics for the Core of the Distributed Ontology Language

Till MOSSAKOWSKI [a,b,1], Christoph LANGE [a] and Oliver KUTZ [a]

[a] *Research Centre on Spatial Cognition, University of Bremen, Germany*
[b] *DFKI GmbH Bremen*

Abstract. The Distributed Ontology Language DOL, which is currently being standardised as ISO WD 17347 within the OntoIOp (Ontology Integration and Interoperability) activity of ISO/TC 37/SC 3, aims at providing a unified framework for (1) ontologies formalised in heterogeneous logics, (2) modular ontologies, (3) links between ontologies, and (4) annotation of ontologies.

A DOL ontology consists of modules formalised in basic ontology languages, such as OWL or Common Logic, which are serialised in the existing syntaxes of these languages. On top of this, DOL provides a meta-level which allows for expressing heterogeneous ontologies and links between ontologies. Such links include (heterogeneous) imports and alignments, conservative extensions, and theory interpretations. This paper focuses on the abstract syntax and semantics of these meta-level constructs. It introduces three alternative semantics for the meta-level, namely *direct*, *translational*, and *collapsed* semantics (the latter is only briefly sketched), and studies their respective pros and cons.

Keywords. ontology languages, logic, heterogeneous formalisation, modular ontologies, syntax, semantics

1. Introduction

OWL is a popular language for ontologies.[2] Yet, the restriction to a decidable description logic often hinders ontology designers from expressing knowledge that cannot (or can only in quite complicated ways) be expressed in a description logic. A practice to deal with this problem is to intersperse OWL ontologies with first-order axioms, e.g. in the case of Bio-ontologies where mereological relations such as parthood are of great importance, though only partly definable in OWL (cf. Section 3). However, these remain informal annotations to inform the human designer, rather than first-class citizens of the ontology with formal semantics and impact on reasoning. One goal of this paper is to equip such heterogeneous ontologies with a precise semantics and proof theory.

---

[1]Corresponding Author: Till Mossakowski, DFKI GmbH Bremen, Enrique-Schmidt-Straße 5, 28359 Bremen, Germany; E-mail: till.mossakowski@dfki.de.

[2]This paper adopts the completely formal position that an ontology is nothing but a formal theory in a given ontology language, and that an ontology language is any logical language that is considered suitable for ontology design by some community.

A variety of languages is used for formalising ontologies. Some of these, such as RDF (mostly used for data), OBO and certain[3] UML class diagrams, can be seen more or less as fragments and notational variants of OWL, while others, like F-logic and Common Logic (CL), clearly go beyond the expressiveness of OWL.

In this paper, we face this diversity not by proposing yet another ontology language that would subsume all the others, but by accepting this pluralism in ontology languages and by formulating means (on a sound and formal semantic basis) to compare and integrate ontologies that are written in different formalisms. This view is a bit different from that of unifying languages such as OWL and CL, which are meant to be "universal" formalisms (for a certain domain/application field), into which everything else can be mapped and represented. While such "universal" formalisms are clearly important and helpful for reducing the diversity of formalisms, it is still a matter of fact that no single formalism will be the Esperanto that is used by everybody. (Below, we will also provide some technical facts supporting this view.) It is therefore important to both accept the existing diversity of formalisms and to provide means of organising their coexistence in a way that enables formal interoperability among ontologies.

2. The Distributed Ontology Language DOL

In this work, we lay the foundation for a distributed ontology language DOL, which will allow users to use their own preferred ontology formalism while becoming interoperable with other formalisms (see [13] for further details). The DOL language is in particular intended to be at the core of a new ISO standardisation effort on ontology interoperability called OntoIOp.[4] At the heart of our approach is a graph of ontology languages and translations. This graph will enable users to *distribute* ontologies in the following ways:

- to relate ontologies that are written in different formalisms. E.g. state that some OWL version of the foundational ontology DOLCE is logically entailed by the (reference) first-order version;
- to re-use ontology modules even if they have been formulated in a different formalism;
- to re-use ontology tools like theorem provers and module extractors along translations between formalisms.

As DOL uses IRIs [6] for globally unique identification, ontologies can furthermore be distributed over the Web.

DOL intends to cover all state-of-the-art basic ontology languages, and to provide a meta level on top of these. This meta level allows for the representation of logically heterogeneous ontologies in the sense that DOL ontologies may comprise of modules written in ontology languages with different underlying logics. Moreover, the DOL meta level constructs allow for links between ontologies such as relative interpretations or conservative extensions.

[3] Those avoiding qualified associations (amounting to identification constraints), n-ary relations (for $n > 2$) and stereotyping.

[4] DOL is currently under standardisation as Working Draft ISO/WD 17347 in ISO/TC 37/SC 3 'Systems to manage terminology, knowledge and content'. See also `http://ontolog.cim3.net/cgi-bin/wiki.pl?OntoIOp`

DOL intends to be an extensible framework for ontology languages. It is intended that any ontology language and any logic whose conformance with DOL has been established can be used with DOL. In particular, we are interested in establishing the conformance of a number of widely used ontology languages and logics as a part of the standard; these include (ordered by increasing complexity) propositional logic (Prop), OWL [1] (with its profiles EL, RL and QL [22]), standard first-order logic with equality ($\mathsf{FOL}^{=}$) and Common Logic (CL) [4][5], as well as translations between these ontology languages. Such translations have been developed in previous research; see e.g. [20].

Note in particular that, among these ontology languages, Common Logic is the most expressive one, and is therefore a target language for translations from all the other languages. Thus, when reasoning about heterogeneous distributed ontologies, one can prima facie translate all participating ontologies to Common Logic. However, note the difference to translating all ontologies to Common Logic in the first place: when, e.g., an OWL ontology has been translated to Common Logic, it is no longer easily amenable to decidable or even tractable reasoning procedures that OWL tools support. Therefore, DOL leaves all ontologies in their original formalisation to take advantage of the optimised automated reasoners for that particular language, and DOL tools should only translate them on demand.

3. An Introductory Example

We use an example from mereology, which lends itself well to heterogeneous formalisation. While mereological relations such as parthood are frequently used in ontologies, many of these ontologies are formalised in languages that are not fully capable of *defining* the mereological notions. For example, mereological relations are used in large biomedical ontologies, which are implemented in the EL profile of OWL for efficiency. OWL is partly capable of defining these relations, whereas more complete definitions require first-order or even second-order logic [14].

Listing 1 shows a heterogeneous mereological ontology. Some parts are written in OWL, others require the greater expressiveness of CL. Moreover, some simple part of the mereology is additionally formulated in Prop. The organisation in a heterogeneous ontology has the advantage that specialised proof tools can be applied to the different parts. More specifically, listing 1 starts with a Prop formalisation of the taxonomy of the categories over which DOLCE [18] defines mereological relations. Although Prop is rarely regarded as an ontology language, this logic is quite popular for formal modelling since consistency and logical consequence can be quite efficiently decided using SAT solvers. In particular, for early detection of modelling errors in an ontology design (especially when using a large number of classes), initial consistency checks and satisfiability of classes can be delegated to a SAT solver for debugging.

We specify a similar ontology in OWL (alternatively we could import the Prop ontology). As this ontology declares classes for all categories that are declared as propositional variables in the Prop ontology satisfying the same disjointness and subsumption relationships, the OWL ontology *interprets* the Prop ontology (we here assume the

[5] In the future, we might add other logics such as IKL [9].

Prop→OWL ontology language translation from Definition 9 below[6]). In the OWL ontology, we additionally introduce parthood properties, which clearly goes beyond the expressivity of Prop. OWL still has quite good automatic reasoning support, but therefore also has its limits: OWL is not capable of defining e.g. the antisymmetry of the isPartOf relation. Overcoming this restriction, we give a full definition of several mereological relations in a CL ontology, which imports and extends the OWL ontology (which is implicitly translated using the default OWL→CL translation). CL extends FOL with second-order style modelling, of which we use the possibility to quantify over predicates here. This allows us to concisely express the restriction of the variables x, y, and z to the same taxonomic category (perdurants, abstract regions, etc.), and it allows us to define a notion of second-order fusion [10].

Listing 1: A heterogeneous ontology for mereology [14]

```
%prefix( :      <http://www.example.org/mereology#>
         owl:   <http://www.w3.org/2002/07/owl#>
         log:   <http://purl.net/dol/logics/>
         trans: <http://purl.net/dol/translations/> )

distributed-ontology Mereology

logic log:Propositional                        %% syntax used: similar to OWL Manchester [11]
ontology Taxonomy =                        %% DOLCE's basic taxonomic information about mereology
  props PT %[ Particular  ]%, PD %[ Perdurant      ]%, T %[ TimeInterval ]%,
        S  %[ SpaceRegion ]%, AR %[ AbstractRegion ]%
  . S ∨ T ∨ AR ∨ PD ⟶ PT %% PT is the top concept
  . S ∧ T  ⟶ ⊥ %% PD, S, T, AR are pairwise disjoint
  . T ∧ AR ⟶ ⊥ %% ...

logic log:OWL                                    %% syntax: OWL Manchester serialization [11]
ontology BasicParthood =                         %% Parthood in OWL DL, as far as expressible
  Class: ParticularCategory SubClassOf: PT                 %% other class declarations omitted
    DisjointUnionOf: S, T, AR, PD                       %% pairwise disjointness more compact
  ObjectProperty: isPartOf       Characteristics: Transitive
  ObjectProperty: isProperPartOf Characteristics: Transitive, Asymmetric  SubPropertyOf: isPartOf
  Class: Atom EquivalentTo: inverse isProperPartOf only owl:Nothing

interpretation TaxonomyToParthood : Taxonomy with logic PropToOWL to BasicParthood

logic log:CommonLogic                        %% syntax: CLIF dialect of Common Logic [4, Annex A]
ontology ClassicalExtensionalParthood =
  BasicParthood then {                      %% import OWL ontology from above, translate it to CL
  . (forall (X) (if (or (= X S) (= X T) (= X AR) (= X PD))
                    (forall (x y z) (if (and (X x) (X y) (X z))
                                        (and                          %% now list all the axioms
      (if (and (isPartOf x y) (isPartOf y x)) (= x y))                               %% antisymmetry
      (iff (overlaps x y) (exists (pt) (and (isPartOf pt x) (isPartOf pt y))))
      (iff (isAtomicPartOf x y) (and (isPartOf x y) (Atom x)))
      (iff (sum z x y) (forall (w) (iff (overlaps w z) (and (overlaps w x) (overlaps w y)))))
      (exists (s) (sum s x y)))))))                                             %% existence of the sum
  . (forall (Set a) (iff (fusion Set a)                 %% definition of fusion
            (forall (b) (iff (overlaps b a) (exists (c) (and (Set c) (overlaps c a))))))) }
```

4. Syntax of DOL

The meta level on top of basic ontologies is specified with its syntax (this section) and semantics (Section 6). Note that the DOL language (and its semantics) as introduced here only covers the core meta logical constructs that will be part of the full DOL language.[7]

[6]As Prop→OWL is a non-default translation, it has to be stated explicitly; cf. the end of Section 5 for an explanation.

[7]Features not covered here for conciseness include e.g. heterogeneous alignments and heterogeneous least upper bounds of ontology collections (i.e. heterogeneous colimits of logical theories). But see [15] for a concrete use of these constructs.

4.1. Distributed Ontologies

A distributed ontology consists of at least one (possibly heterogeneous) **ontology**, plus, optionally, **interpretations** between its participating ontologies. More specifically, a distributed ontology consists of a name, followed by a list of `LOGIC-SECTION`s. A `LOGIC-SECTION` selects a specific ontology language (actually the underlying *logic*) that is used to interpret the subsequent `DIST-ONTO-ITEM`s. A `DIST-ONTO-ITEM` is either an **ontology definition** (`ONTO-DEFN`), or an **interpretation** between ontologies (`INTPR-DEFN`). Alternatively, a distributed ontology can also be the verbatim inclusion of an ontology written in a specific ontology language, using that language's structuring construct instead of DOL's (`ONTO-IN-SPECIFIC-LANGUAGE`).

```
DIST-ONTO-DEFN ::= distributed-ontology DIST-ONTO-NAME LOGIC-SECTION*
                 | ONTO-IN-SPECIFIC-LANGUAGE %% logic-specific
LOGIC-SECTION  ::= logic LOGIC-REF DIST-ONTO-ITEM*
DIST-ONTO-ITEM ::= ONTO-DEFN | INTPR-DEFN
```

4.2. Heterogeneous Ontologies

An **ontology** O can be, among other cases not covered here, one of the following:

- a **basic ontology** $\langle\Sigma,\Delta\rangle$ written in some ontology language. For simplicity, we assume that it is directly given by a signature Σ and a set of axioms Δ, although in reality, basic ontologies are written in a specific concrete syntax, from which a signature and a set of axioms can then be extracted. Often, one can even choose among different concrete syntaxes[8];
- a **translation** "O with logic ρ" of an ontology O along an ontology language translation ρ;
- an **extension** O then $CS?$ $\langle\Sigma,\Delta\rangle$ of an ontology O by another basic ontology $\langle\Sigma,\Delta\rangle$; the extension can be marked as conservative (CS; model-conservative or consequence-conservative)
- a **reference** to an ontology existing on the Web,
- an ontology O qualified with the ontology language (logic) used to express it.

$$O ::= \langle\Sigma,\Delta\rangle \,|\, O \text{ with logic } \rho \,|\, O \text{ then } CS?\ \langle\Sigma,\Delta\rangle \,|\, \texttt{ONTO-REF} \,|\, \text{logic } \texttt{LOGIC-REF}\ O$$

$$CS ::= \%mcons \,|\, \%ccons$$

An **ontology definition** `ONTO-DEFN` names an ontology O. It can be optionally marked as consistent, using CS. An ontology language **translation** ρ is either specified by its name, or it is inferred as the default translation between a given source and target ontology language. A **link** provides a connection between two ontologies. Finally, DOL uses IRIs (Internationalised Resource Identifiers [6]), a generalisation of URIs, for naming.

[8] OWL calls them *serialisations*; CL has different *dialects*, which also have different syntaxes. A thorough treatment of this issue will be given in the OntoIOp standard.

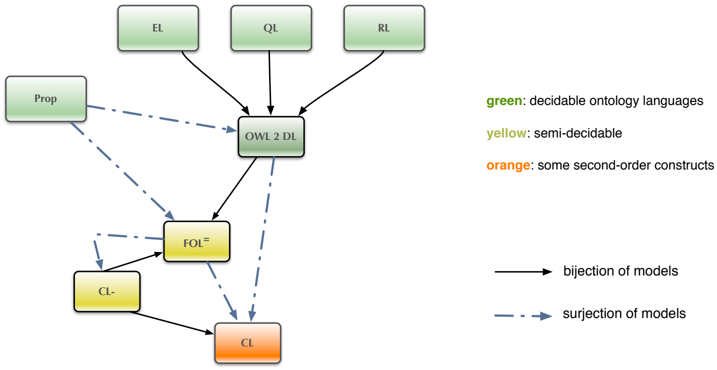

Figure 1. The logic translation graph for basic ontology languages discussed in this paper.

```
ONTO-DEFN  ::= ontology ONTO-NAME = O end
INTPR-DEFN ::= interpretation INTPR-NAME : O to O end
DIST-ONTO-NAME | ONTO-NAME | ONTO-REF | LOGIC-REF | INTPR-NAME ::= IRI
IRI ::= see RFC 3987 [6, Section 2.2]
```

5. The Logic Graph

We now define the graph of logics and logic translations that will be used in DOL. Each logic will come with notions of *sentence* and *model*, and a relation $\models$ of *satisfaction* between these. We will tacitly extend $\models$ to sets of models and sets of sentences, meaning that all models satisfy all sentences. Moreover, DOL provides means for talking about conservative extensions. An extension of a theory is a super-theory that does not introduce new properties (over the vocabulary of the old theory). Hence, we also need to provide a notion of *signature* (vocabulary) and of inclusion between signatures; that is, signatures are partially ordered. Signature inclusion is usually defined component-wise, that is, each component of the smaller signature must be included in the component of the larger signature. Given signatures $\Sigma_1 \leq \Sigma_2$, we assume that all Σ_1-sentences are also Σ_2-sentences; moreover, we assume that each Σ_2 model M_2 has a *reduct* $M_2|_{\Sigma_1}$ to a Σ_1-model (M_2 is then called an *expansion* of $M_2|_{\Sigma_1}$). Usually, this reduct is given just by deleting the components for $\Sigma_2 \setminus \Sigma_1$ from Σ_2; we will hence not discuss reducts in specific logics below. For each logic, it is easy to show that $M_2 \models \varphi$ iff $M_2|_{\Sigma_1} \models \varphi$, that is, satisfaction is invariant under reduct. Finally, we assume that there is a union operation on signatures.

Definition 1. Propositional Logic. Signatures in Prop are just sets Σ (of propositional symbols) as signatures, and signature inclusion is just set inclusion. A Σ-model M is a mapping from Σ to $\{true, false\}$. Σ-sentences are built from Σ with the usual propositional connectives. Finally, satisfaction of a sentence in a model is defined by the standard truth-table semantics.

Definition 2. OWL **2 DL.** OWL 2 DL is the description logic (DL) based fragment of the web ontology language OWL 2. We start with the simple description logic $\mathcal{ALC}$, and then proceed to the more complex description logic $\mathcal{SROIQ}$ which is underlying OWL 2 DL. Signatures of the description logic $\mathcal{ALC}$ consist of a set $\mathcal{A}$ of atomic concepts, a set $\mathcal{R}$ of roles and a set $\mathcal{I}$ of individual constants. Models are first-order structures $I = (\Delta^I, .^I)$ with universe Δ^I that interpret concepts as unary and roles as binary predicates (using $.^I$). Sentences are subsumption relations $C_1 \sqsubseteq C_2$ between concepts, where concepts follow the grammar

$$C ::= \mathcal{A} \,|\, \top \,|\, \bot \,|\, C_1 \sqcup C_2 \,|\, C_1 \sqcap C_2 \,|\, \neg C \,|\, \forall R.C \,|\, \exists R.C$$

These kind of sentences are also called TBox sentences. Sentences can also be ABox sentences, which are membership assertions of individuals in concepts (written $a : C$ for $a \in \mathcal{I}$) or pairs of individuals in roles (written $R(a,b)$ for $a, b \in \mathcal{I}, R \in \mathcal{R}$). Satisfaction is the standard satisfaction of description logics.

The logic $\mathcal{SROIQ}$ [12], which is the logical core of the Web Ontology Language OWL 2 DL[9], extends $\mathcal{ALC}$ with the following constructs: (i) complex role inclusions such as $R \circ S \sqsubseteq S$ as well as simple role hierarchies such as $R \sqsubseteq S$, assertions for symmetric, transitive, reflexive, asymmetric and disjoint roles (called RBox sentences, denoted by $\mathcal{SR}$), as well as the construct $\exists R.\mathsf{Self}$ (collecting the set of 'R-reflexive points'); (ii) nominals, i.e. concepts of the form $\{a\}$, where $a \in \mathcal{I}$ (denoted by $\mathcal{O}$); (iii) inverse roles (denoted by $\mathcal{I}$); qualified and unqualified number restrictions ($\mathcal{Q}$). For details on the rather complex grammatical restrictions for $\mathcal{SROIQ}$ (e.g. regular role inclusions, simple roles) compare [12].

OWL *profiles* are syntactic restrictions of OWL 2 DL that support specific modelling and reasoning tasks [22], and which are accordingly based on DLs with appropriate computational properties. Specifically, OWL 2 EL is designed for ontologies containing large numbers of concepts or relations, OWL 2 QL to support query answering over large amounts of data, and OWL 2 RL to support scalable reasoning using rule languages (EL, QL, and RL for short) .

We sketch the logic $\mathcal{EL}$ which is underlying the EL profile.[10] $\mathcal{EL}$ is a syntactic restriction of $\mathcal{ALC}$ to existential restriction, concept intersection, and the top concept:

$$C ::= \mathcal{A} \,|\, \top \,|\, C_1 \sqcap C_2 \,|\, \exists R.C$$

Note that $\mathcal{EL}$ does not have disjunction or negation, and is therefore a sub-Boolean logic.

Definition 3. Untyped First-order Logic $\mathsf{FOL}^=$**.** Signatures are first-order signatures, consisting of a set of function symbols with arities, and a set of predicate symbols with arities. Models are first-order structures, and sentences are first-order formulae. Satisfaction is the usual satisfaction of a first-order sentence in a first-order structure.

Definition 4. Common Logic. A common logic signature Σ (called vocabulary in Common Logic terminology) consists of a set of names, with a subset called the set of discourse names, and a set of sequence markers. An inclusion of signatures needs to fulfil the requirement that a name is a discourse name in the smaller signature if and only if it

[9] See also `http://www.w3.org/TR/owl2-overview/`

[10] To be exact, EL adds various 'harmless' expressive means and syntactic sugar to $\mathcal{EL}$ resulting in the DL $\mathcal{EL}++$ [2]; for further details see also [22].

is one in the larger signature. A Σ-model consists of a set UR, the universe of reference, with a non-empty subset $UD \subseteq UR$, the universe of discourse, and four mappings:

- *rel* from UR to subsets of $UD^* = \{< x_1, \ldots, x_n > | x_1, \ldots, x_n \in UD\}$ (i.e., the set of finite sequences of elements of UD);
- *fun* from UR to total functions from UD^* into UD;
- *int* from names in Σ to UR, such that $int(v)$ is in UD if and only if v is a discourse name;
- *seq* from sequence markers in Σ to UD^*.

A Σ-sentence is a first-order sentence, where predications and function applications are written in a higher-order like syntax: $t(s)$. Here, t is an arbitrary term, and s is a sequence term, which can be a sequence of terms $t_1 \ldots t_n$, or a sequence marker. A predication $t(s)$ is interpreted by evaluating the term t, mapping it to a relation using *rel*, and then asking whether the sequence given by the interpretation s is in this relation. Similarly, a function application $t(s)$ is interpreted using *fun*. Otherwise, interpretation of terms and formulae is as in first-order logic. A further difference is the presence of sequence terms (namely sequence markers and juxtapositions of terms), which denote sequences in UD^*, with term juxtaposition interpreted by sequence concatenation. Note that sequences are essentially a non-first-order feature that can be expressed in second-order logic. For details, see [4]. We call the restriction of CL to sentence without sequence markers CL $^-$.

We now define a number of logic translations[11], which consist of three components: a signature translation Φ, which is expected to map signature extensions to signature extensions, a sentence translation α and a model translation β (where models are translated in the reverse direction). Generally, we allow that models are translated to sets of models; if models are translated to single models, we call the model translation single-valued. Each translation enjoys the following representation condition:

$$\beta(M) \models \varphi \text{ iff } M \models \alpha(\varphi),$$

where φ is a sentence in the source of the translation, and M is a model in the target of the translation. Moreover, we require that each model translation is surjective, is compatible with reduct, i.e. $\beta(M'|_{\Phi(\Sigma)}) = \beta(M')|_\Sigma$ for $\Sigma \leq \Sigma'$ and $M' \in Mod(\Phi(\Sigma'))$. All these properties hold for all the translations below and are generally straightforward to show. Surjectivity of model translation leads to the pleasant property that logical consequence is faithfully reflected along translations:

$$\Gamma \models \varphi \text{ iff } \alpha(\Gamma) \models \alpha(\varphi)$$

A further property is needed for compatibility of the different semantics, see Theorem 13 below. We call a translation *weakly exact*, if it is single-valued (that is, each $\beta(M)$ is a singleton, and, by abuse of notation, we will write $\beta(M)$ also for its unique element), and moreover for each $M' \in Mod(\Sigma')$ and $M_1 \in Mod(\Phi(\Sigma))$ with $\beta(M_1) = M'|_\Sigma$, there is a model $M'_1 \in Mod(\Phi(\Sigma'))$ with $M'_1|_{\Phi(\Sigma)} = M_1$ and $\beta(M'_1) = M'$.

[11] Further translations arise through composition, which is defined component wise.

Definition 5. Translation from Prop to $\mathsf{FOL}^=$. The mapping Prop $\rightarrow$ $\mathsf{FOL}^=$ translates propositional variables to nullary predicates. This naturally extends to sentences. The model translation forgets the universe. Due to this, the model translation is only surjective[12] but not bijective.

Definition 6. Translation from EL to OWL. Since EL is a syntactic restriction of OWL, there is an obvious inclusion from EL into OWL. The model translation is bijective.

Definition 7. Translation from OWL to $\mathsf{FOL}^=$. This is the well-known standard translation into the two-variable fragment of untyped first-order logic, mapping concepts to unary predicates and roles to binary predicates [23]. The model translation is bijective.

Definition 8. Translation from $\mathsf{FOL}^=$ to CL. The signature translation maps constants, function symbols and predicates to names. Sentences are left untouched. From a CL-model, it is possible to extract a $\mathsf{FOL}^=$-model by restricting functions and predicates to those sequences that have the length of the arity of the symbol. Due to this forgetting of function and predicate on other sequences, the model translation is only surjective, but not bijective.

Definition 9. Translation from Prop to OWL. Each propositional variable p in a signature is mapped to an atomic OWL class P. This mapping naturally extends to a mapping from complex propositions to complex OWL classes by replacing the respective Boolean operations. A propositional sentence is mapped to equality of the corresponding class with the top class *Thing*. An OWL model $I = (\Delta^I, .^I)$ is translated to a set $\{I_a \mid a \in \Delta^I\}$ of propositional models, one for each individual $a \in \Delta^I$, mapping each propositional variable p to the truth value of $a \in P^I$, i.e. $I_a(p) \iff a \in P^I$. The model translation is no longer single-valued, but it is still surjective in the sense that all sets of Prop-models are in the image of the model translation.

Definition 10. Translation from OWL to CL. We use the translation defined by Pat Hayes, see `http://www.ihmc.us/users/phayes/CL/SW2SCL.html`. It is similar to FOL$\rightarrow$CL $\circ$ OWL$\rightarrow$FOL, but differs in an important respect. Namely, OWL notions like subclass relationship or inverse roles are axiomatised in CL and can be reused. In particular, theorems can be proven for this axiomatisation once and for all, and an intelligent theory management mechanism could automatically import these theorems to all the translations.

Definition 11. Translation from CL $^-$to FOL $^=$. Christopher Menzel [19] provides a coding from CL $^-$to FOL $^=$ that uses explicit $\mathtt{Holds}_n$ predicates and $\mathtt{App}_n$ functions. This coding could be used for interfacing CL with FOL $^=$ provers and model finders.

We mark all translations except the one from Prop to OWL and the one from OWL to CL as *default translations*. The default translation from logic L_1 to L_2 (if existing) will be denoted by $\mathit{default}(L_1, L_2)$. We further will use any composition of default translations as default translations as well. Note that the default translation from Prop to $\mathsf{FOL}^=$ is different from the composition of the (non-default) translation of Prop to OWL with the (default) translation of OWL to $\mathsf{FOL}^=$. Similarly, FOL$\rightarrow$CL $\circ$ OWL$\rightarrow$FOL $\neq$ OWL$\rightarrow$CL.

[12] Surjective in the sense that for each Prop-model, there is a $\mathsf{FOL}^=$-model being translated to this Prop-model, not in the sense that all sets of Prop-models are in the image of the translation.

6. Semantics of DOL

We pursue a threefold approach of assigning a semantics to the abstract syntax introduced in Section 4:

Direct Model-Theoretic Semantics: On the level of basic ontologies, this semantics reuses the existing semantics of Prop, OWL, $\mathsf{FOL}^=$ and CL, as well as translations between these logics, as introduced in Section 5. The semantics of the meta level is specified in semi-formal mathematical textbook style.

Translational Semantics: The semantics of Common Logic is employed for all basic ontology languages, taking advantage of the fact that Common Logic is a common translation target for all basic ontology languages supported so far, as shown in Section 5. In detail, the translational semantics first translates the abstract syntax of meta$(\mathsf{Prop}, \mathsf{EL}, \ldots)$ into the abstract syntax of meta(CL).[13] The latter is interpreted in terms of the existing Common Logic semantics, while the semantics of the meta language is still specified semi-formally, as in the case of the direct semantics.

Collapsed Semantics: The collapsed semantics extends the translational semantics to a semantics that is fully given specified in Common Logic. It further translates the abstract syntax meta(CL) to Common Logic, and then reuses the semantics of Common Logic, without employing a separate semantics for the meta language. Here, the meta and object levels are collapsed into Common Logic, but may still be distinguished by a closer look into the Common Logic theory.

The model-theoretic nature of the semantics ensures a better representation of the model theory than a theory-level semantics would do. In particular, Theorem 13 ensures that models classes of logical theories represented in Common Logic can be recovered through a model translation. This is of particular importance when studying model-theoretic properties like finite model or tree model properties.

6.1. Direct Semantics

The direct semantics relies on the logics and logic translations as defined in Section 5. Many parts of the semantics use a logic L (the *current logic*) as a parameter. For example, one and the same ontology can be regarded as an EL or an OWL ontology, depending on L. Moreover, the semantics of ontologies generally depends on a global environment Γ mapping `IRI`s to (semantics of) ontologies. In each logic, we denote the empty signature by $\emptyset$ and its model class by $\mathcal{M}_\emptyset$.

We now come to the semantics. A sequence of ontology definitions is just unfolded:

$$\begin{gathered} sem(\Gamma, \texttt{dist-onto-defn DIST-ONTO-NAME } LS_1, \ldots LS_n) = \\ sem(\ldots sem(sem(\Gamma, LS_1), LS_2), \ldots, LS_n) \end{gathered}$$

The selection of a specific logic is propagated to all subsequent ontology definitions:

$$\begin{gathered} sem(\Gamma, \texttt{logic-select LOGIC-REF } DOI_1 \ldots DOI_n) = \\ sem(\ldots sem(sem(\Gamma, L, DOI_1), L, DOI_2), \ldots, L, DOI_n) \text{ where } L = sem(\texttt{LOGIC-REF}) \end{gathered}$$

[13] I.e. meta(CL) is the homogeneous restriction of DOL to CL, but comprising the same meta level constructs.

In the context of a global environment Γ and the current logic L, an ontology O is interpreted as a signature $\Sigma = sig(\Gamma, L, O)$ in some logic $L' = logic(\Gamma, L, O)$ and a class of models $\mathcal{M} = Mod(\Gamma, L, O)$ over that signature. We combine this into

$$sem(\Gamma, L, O) = (logic(\Gamma, L, O), sig(\Gamma, L, O), Mod(\Gamma, L, O)).$$

The following table specifies the semantics of ontologies. The semantics of a basic ontology is given by its signature and the class of model satisfying its axioms. The semantics of the translation of an ontology along a logic translation uses the signature and model translation components of the logic translation[14]. A reference to a named ontology is looked up in the global environment and, if needed, translated to the current logic (this kind of implicit coercion is used in Listing 1 above when including an OWL ontology into a CL ontology). Finally, a logic qualification replaces the current logic with a new one.

O'	$sem(\Gamma, L, O') = \ldots$
$\langle \Sigma, \Delta \rangle$	$(L, \Sigma, \{M \in Mod(\Sigma) \mid M \models \Delta\})$
O with logic ρ	Let $\Sigma = sig(\Gamma, L, O)$ and $\rho = (\Phi, \alpha, \beta) : L_1 \to L_2$. Then $logic(\Gamma, L, O') = L_2$, $sig(\Gamma, L, O') = \Phi(\Sigma)$, and $Mod(\Gamma, L, O') = \{M \in Mod(\Phi(\Sigma)) \mid \beta(M) \subseteq Mod(\Gamma, L, O)\}$
O then CS? $\langle \Sigma', \Delta' \rangle$	Let $\Sigma = sig(\Gamma, L, O)$. Then $sig(\Gamma, L, O') = \Sigma \cup \Sigma'$ $Mod(\Gamma, L, O') =$ $\{M' \in Mod(\Sigma \cup \Sigma') \mid M' \models \Delta'$ and $M'\|_\Sigma \in Mod(\Gamma, L, O)\}$
`ONTO-REF`	$(L, \Phi(\Sigma), \{M \in Mod(\Phi(\Sigma)) \mid \beta(M) \subseteq \mathcal{M}\}$ where $\Gamma($`ONTO-REF`$) = (L_1, \Sigma, \mathcal{M})$ and $(\Phi, \alpha, \beta) : L_1 \to L$ is the default translation
logic `LOGIC-REF` O	$sem(\Gamma,$ `LOGIC-REF`$, O)$

For an extension, if *CS* is *%mcons* (model-conservative), the semantics is only defined if each model in $Mod(\Gamma, L, O')$ can be expanded to a model in $Mod(\Gamma, L, O)$. If *CS* is *%ccons* (consequence-conservative), the semantics is only defined if for each Σ-sentence φ, $Mod(\Gamma, L, O) \models \varphi$ implies $Mod(\Gamma, L, O') \models \varphi$.

Proposition 12. Model-conservativity is stronger than consequence-conservativity.

Proof. Assume that each model in $Mod(\Gamma, L, O')$ can be expanded to a model in $Mod(\Gamma, L, O)$. Further assume $Mod(\Gamma, L, O) \models \varphi$. If $M \in Mod(\Gamma, L, O')$, then M can be expanded to $M' \in Mod(\Gamma, L, O)$. By assumption, $M' \models \varphi$. Since satisfaction is invariant under reduct, also $M \models \varphi$. □

An ontology definition extends the global environment:

$$sem(\Gamma, L, \texttt{ontology ONTO-NAME} = O) = \Gamma[\texttt{ONTO-NAME} \mapsto sem(\Gamma, L, O)]$$

We now come to the semantics of relative interpretations, which is formulated in terms of model class inclusion, since our semantics is model-theoretic.

[14] A theory-level semantics would use the sentence translation instead of the model translation.

$$sem(\Gamma, L, \texttt{interpretation INTPR-NAME : } O_1 \texttt{ to } O_2\texttt{))} =$$
$$\Gamma[\texttt{INTPR-NAME} \mapsto (\Sigma_1, \Sigma_2)] \text{ where } (L_1, \Sigma_1, \mathcal{M}_1) = sem(\Gamma, L, O_1),$$
$$(L_2, \Sigma_2, \mathcal{M}_2) = sem(\Gamma, L, O_2), \text{ and the semantics is defined only if } \beta(\mathcal{M}_2|_{\Sigma_1}) \subseteq \mathcal{M}_1,$$
$$\text{where } (\Phi, \alpha, \beta) \text{ is the default translation from } L_1 \text{ to } L_2.$$

We gave a direct semantics $sem(\Gamma, L, O) = (logic(\Gamma, L, O), sig(\Gamma, L, O), Mod(\Gamma, L, O))$ to any heterogeneous ontology O, in the context of a current logic L[15] and a global environment Γ specifying how references to external ontologies are resolved. With this semantics, one can define many standard logical notions in a straightforward way. For example, a heterogeneous ontology is **satisfiable** if it has a nonempty model class. A sentence φ is a **logical consequence** of a heterogeneous ontology O in context Γ and L, written $\Gamma, L, O \models \varphi$, if φ is a sentence in the logic $logic(\Gamma, L, O)$, and each model in $Mod(\Gamma, L, O)$ satisfies φ.

6.2. Translational Semantics

The translational semantics uses Common Logic as a foundational framework for the distributed ontology language DOL, similar to what set theory provides for general mathematical theories. This semantics assumes that each involved ontology language is mapped to CL by a weakly exact translation. The semantics is defined by first translating a heterogeneous ontology to CL, and then using the direct semantics for the result.

Note that since the result of translating a DOL ontology entirely to CL is homogeneous, the clause for logic translation of the direct semantics will not be used. In Section 5, we have defined a number of default logic translation. Using these and compositions of these, every logic can be mapped to Common Logic, while the meta constructs like interpretations stay the same.

We define the syntactic translation CL_ρ of DOL ontologies, depending on a logic translation $\rho : L \to \mathsf{CL}$, to Common Logic below. (The translations of the other syntactic categories are straightforward.)

$$CL_\rho(\langle \Sigma, \Delta \rangle) = \langle \Phi(\Sigma), \alpha(\Delta) \rangle, \text{ where } \rho = (\Phi, \alpha, \beta)$$
$$CL_\rho(O \texttt{ with logic } \rho') = CL_{\rho \circ \rho'}(O)$$
$$CL_\rho(O \texttt{ then } CS\ \langle \Sigma, \Delta \rangle) = CL_\rho(O) \texttt{ then } CS\ CL_\rho(\langle \Sigma, \Delta \rangle)$$
$$CL_\rho(\texttt{ONTO-REF}) = \texttt{ONTO-REF}$$
$$CL_\rho(\texttt{logic LOGIC-REF } O) = CL_{default(\texttt{LOGIC-REF}, \mathsf{CL})}(O)$$

6.3. Collapsed Semantics

The collapsed semantics requires the representation of the meta level within CL. For this purpose, the model-level semantics introduced in the previous section should be complemented by a theory-level semantics: a distributed ontology then denotes a basic theory in some logic (which amounts to flattening out all structure), plus some conditions for conservativity and relative interpretations. For each logic, one needs to axiomatise a specific partial order of signatures in CL, plus a set of sentences equipped with a logical consequence relation. In order to avoid the formalisation of models and the satisfaction relation (which would require the inclusion of a set theory like ZFC), a sound and com-

[15] Note that L need not coincide with $logic(\Gamma, L, O)$, because DOL contains constructs for selecting a new logic and also for translating an ontology to a different logic.

plete calculus is axiomatised for each logic. For each logic translation, the signature and sentence translations need to be axiomatised. We require that this axiomatisation is done in such a way that the resulting semantics is compatible with the translational semantics. Although this formalisation is doable in principle, we refrain form providing the (massive) details.

7. Relations Among the Different Semantics

We now show that the translational semantics is compatible with the direct semantics. Two global environments Γ and Γ_{CL}, where the latter one involves the logic CL only, are said to be **compatible**, if for each ontology name `ON`,

$$\Phi(\Sigma) = \Sigma_{\mathsf{CL}} \text{ and } \mathcal{M} = \beta(\mathcal{M}_{\mathsf{CL}})$$

where $\Gamma[\mathtt{ON}] = (L, \Sigma, \mathcal{M})$ and $\Gamma_{CL}[\mathtt{ON}] = (\mathsf{CL}, \Sigma_{CL}, \mathcal{M}_{CL})$ and $\rho = (\Phi, \alpha, \beta)$ is the default translation from L to CL.

Theorem 13 (Compatibility of direct semantics and translational semantics)**.** Let Γ and Γ_{CL} be compatible global environments, and let `ONTO` be an ontology not involving conservative extensions written in logic L_1. Then

$$\Phi(\Sigma) = \Sigma_{\mathsf{CL}} \text{ and } \mathcal{M} = \beta(\mathcal{M}_{\mathsf{CL}})$$

where $sem(\Gamma, L_1, \mathtt{ONTO}) = (L, \Sigma, \mathcal{M})$, $sem(\Gamma_{CL}, \mathsf{CL}, CL_\rho(\mathtt{ONTO})) = (\mathsf{CL}, \Sigma_{\mathsf{CL}}, \mathcal{M}_{\mathsf{CL}})$, and $\rho = (\Phi, \alpha, \beta)$ is the default translation from L to CL.

Proof. The proof proceeds by induction over the structure of `ONTO`. For basic ontologies, we use the representation condition for logic translations. For logic translations, we use a simple calculation using composition of translations. For references to named ontologies, we use compatibility of the global environments. Qualifications with a logic simply use the induction hypothesis. The only more involved case is that of extensions. Given an ontology $O' = O \; \mathtt{then} \; \langle \Sigma', \Delta' \rangle$, let $sem(L_1, \Gamma, O) = (L, \Sigma, \mathcal{M})$ and $sem(\mathsf{CL}, \Gamma_{\mathsf{CL}}, CL_\rho(O)) = (\mathsf{CL}, \Sigma_{\mathsf{CL}}, \mathcal{M}_{\mathsf{CL}})$. By the induction hypothesis, we know that $\Phi(\Sigma) = \Sigma_{\mathsf{CL}}$ and $\beta(\mathcal{M}_{\mathsf{CL}}) = \mathcal{M}$. Let $sem(L_1, \Gamma, O') = (L, \Sigma \cup \Sigma', \mathcal{M}')$ and $sem(\mathsf{CL}, \Gamma_{\mathsf{CL}}, CL_\rho(O')) = (\mathsf{CL}, \Sigma'_{\mathsf{CL}}, \mathcal{M}'_{\mathsf{CL}})$. Now clearly, $\Sigma'_{\mathsf{CL}} = \Phi(\Sigma \cup \Sigma')$. We need to show that $\beta(\mathcal{M}'_{\mathsf{CL}}) = \mathcal{M}'$. We first show $\beta(\mathcal{M}'_{\mathsf{CL}}) \subseteq \mathcal{M}'$. Assume that $M' \in \mathcal{M}'_{\mathsf{CL}}$, that is, $M' \models \alpha(\Delta')$ and $M'|_{\Sigma_{\mathsf{CL}}} \in \mathcal{M}_{\mathsf{CL}}$. By the representation condition, $\beta(M') \models \Delta'$, and since model translation is compatible with reduct, $\beta(M')|_\Sigma = \beta(M'|_{\Sigma_{\mathsf{CL}}}) \in \mathcal{M}$. Hence, $\beta(M') \in \mathcal{M}'$. Now let us show that $\mathcal{M}' \subseteq \beta(\mathcal{M}'_{\mathsf{CL}})$. Assume that $M \in \mathcal{M}'$, i.e. $M \models \Delta'$ and $M|_\Sigma \in \mathcal{M}$. Then there is some $M' \in \mathcal{M}_{\mathsf{CL}}$ with $\beta(M') = M|_\Sigma$. By weak exactness, there is some M'' with $M''|_{\Sigma_{\mathsf{CL}}} = M'$ and $\beta(M'') = M$. By the representation condition, $M'' \models \alpha(\Delta')$. Since also $M''|_{\Sigma_{\mathsf{CL}}} = M' \in \mathcal{M}_{\mathsf{CL}}$, altogether $M'' \in \mathcal{M}'_{\mathsf{CL}}$, hence $M \in \beta(\mathcal{M}'_{\mathsf{CL}})$. □

Theorem 13 naturally extends to the semantics of distributed ontologies. The theorem shows that we can extract Prop-, EL-, OWL- or FOL-models out of the CL-models provided by the translational semantics. However, note the model translations are generally not bijective. This means that one and the same model can be represented by several CL models.

While theorem 13 provides a good compatibility of the first two semantics, there are some differences. One obvious difference is that the translational semantics requires a translation of all involved languages to CL, while the direct semantics works also without such translations. More subtle differences emerge from meta-theoretical properties of logics as well as reasoning tasks: some of these depend on syntactic features of the logics, and are therefore not straightforwardly preserved under translations in more expressive formalisms. We discuss next in some detail two examples, conservative extensions and the computation of least common subsumers.

Example 14 (Conservative extensions)**.** Conservative extensions, both in the consequence-theoretic and the model-theoretic variant, play an important role, in particular concerning ontology module extraction, see e.g. [17]. Consider a large ontology like SNOMED CT [24]. For a particular application, typically only a small portion, i.e. a subtheory, of the ontology is needed. However, this subtheory should contain all logical information about its signature that is implied by the whole ontology. This requirement is precisely that of conservative extension. Hence, a **module** is a subtheory of an ontology such that the ontology is a conservative extension of the module. Consider the following EL theory about lectures and their subjects [16]:[16]

$$\begin{aligned} \mathsf{Lecture} &\sqsubseteq \exists \mathsf{has\_subject}.\mathsf{Subject} \sqcap \exists \mathsf{given\_by}.\mathsf{Lecturer} \\ \mathsf{Intro\_AI} &\sqsubseteq \mathsf{Lecture} \end{aligned}$$

This theory is extended as follows:

$$\begin{aligned} \mathsf{Intro\_AI} &\sqsubseteq \exists \mathsf{has\_subject}.\mathsf{Logic} \\ \mathsf{Intro\_AI} &\sqsubseteq \exists \mathsf{has\_subject}.\mathsf{NeuralNetworks} \\ \mathsf{Logic} \sqcap \mathsf{NeuralNetworks} &\sqsubseteq \bot \end{aligned}$$

Now this extended theory logically implies that $\mathsf{Intro\_AI} \sqsubseteq \geq 2\, \mathsf{has\_subject}$; this follows since Logic and NeuralNetworks are disjoint and both related via has_subject to Intro_AI. Hence, in OWL, the larger theory is not a consequence-theoretic conservative extension of the smaller one, because $\mathsf{Intro\_AI} \sqsubseteq \geq 2\, \mathsf{has\_subject}$ is a sentence in the signature of the smaller theory that follows from the larger theory, but not from the smaller one. But in EL, such a sentence does not exist. In particular, the number restriction $\geq 2\, \mathsf{has\_subject}$ cannot be expressed in EL.

Example 15 (Least common subsumers)**.** Computing the least common subsumer (lcs) of a set of concept descriptions is an important non-standard reasoning task in DLs, used in particular in the bottom-up construction of ontologies. For a DL L and (complex) concepts $C_1, \ldots C_n$ expressed in L, the lcs for these concepts is the least (under concept subsumption) concept C expressible in L such that $C_i \sqsubseteq C$ for all i. Depending on the DL under consideration, the lcs need not exist but if it does is always unique up to equivalence. In [3] it was shown that for the DL $\mathcal{EL}$, the lcs always exists and can be computed in polynomial time. Note that for DLs containing disjunction, the operation is trivialised. In particular, an lcs for $\mathcal{EL}$ is typically not an lcs for the same concepts but expressed within a more expressive DL L' extending $\mathcal{EL}$.

[16] Note that the $\bot$ concept is part of the $\mathcal{EL}$ ++ extension of $\mathcal{EL}$ and so available in EL.

These examples show that the translational semantics differs from the direct semantics w.r.t reasoning tasks that are syntax sensitive. Concerning conservative extensions, while the direct semantics would recognise the above extension as consequence-theoretically conservative (because it directly works with EL), the translational semantics would not, because after translating to CL, the conservative extension property is lost. In order to keep this property, one needs to introduce a more fine-grained notion of consequence-theoretic conservative extension in CL, which allows the specification of a sublogic that is used for interpreting the conservative extension.

Similarly, the property of being an lcs for a set of concepts is lost under the translational semantics. Note also that proof support is more easily handled under the direct semantics, as for instance tools for computing the lcs work directly (and syntax-based) with e.g. the EL logic [5]. Related properties studied in the OWL community, such as Beth definability [25] and Craig interpolation [17], are similarly not preserved in general under logic translations.

8. Conclusion

We have presented a distributed heterogeneous ontology language with three different semantics, which (except from the semantics of conservative extensions) are compatible with each other. While the direct semantics stays close to the semantics of the individual ontology languages such as OWL and CL, the translational semantics is based on a mapping to CL, such that only knowledge of CL is required to understand this semantics. The collapsed semantics formalises also the meta theory in CL, and thus makes the meta level itself amenable to computer-assisted theorem proving and verification.

A first application of DOL is the (indeed homogeneous) COLORE repository [8], containing more than 400 theories written in CL. Currently, the relations between these theories are stated informally and proven manually. With DOL, these relations can be stated formally, and in the near future, we expect the Heterogeneous Tool Set Hets to be able to deal with them, using theorem provers and model finders for the involved logics for (dis-)proving logical facts.

Future work will include more logics and translations, like description logics with features like transitive closure, RuleML and RIF, logics with datatypes, higher-order logics, paraconsistent logics, many-valued logics, logics of uncertainty and probabilistic logics. The direct semantics has a clear advantage here, because these logics can be directly included in the logic graph, while the translational semantics first requires their translation to Common Logic, which in some cases will be quite involved and/or impossible. In the latter case, inclusion of such logics would require a revision of the CL standard.

Another direction of future work will be the generalisation of signature inclusions to signature translations, using the theory of institutions [7,21]. With this, logically heterogeneous ontology alignments as studied in [15] can be covered as well.

Acknowledgements: We would like to thank the OntoIOp working group within ISO/TC 37/SC 3 for providing valuable feedback, in particular Michael Grüninger, Pat Hayes, Maria Keet, Chris Menzel, and John Sowa. This work has been supported by the DFG-funded Research Centre on Spatial Cognition (SFB/TR 8), project I1-[OntoSpace].

References

[1] OWL 2 Web Ontology Language: Document Overview. Recommendation, W3C, October 2009.

[2] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ Envelope Further. In K. Clark and P. F. Patel-Schneider, editors, *Proc. of the OWLED 2008 DC Workshop*. CEUR-WS, 2008.

[3] F. Baader, R. Küsters, and R. Molitor. Computing Least Common Subsumers in Description Logics with Existential Restrictions. In *16th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 96–101, 1999.

[4] Information technology—Common Logic (CL): a framework for a family of logic-based languages. Technical Report 24707:2007, ISO/IEC, 2007. `http://iso-commonlogic.org`.

[5] F. M. Donini, S. Colucci, T. Di Noia, and E. Di Sciascio. A tableaux-based method for computing least common subsumers for expressive description logics. In C. Boutilier, editor, *IJCAI*, pages 739–745, 2009.

[6] M. Dürst and M. Suignard. Internationalized resource identifiers (IRIs). RFC 3987, IETF, 2005.

[7] J. A. Goguen and R. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39:95–146, 1992.

[8] M. Grüninger. COLORE: Common logic ontology repository. `http://code.google.com/p/colore/`.

[9] P. J. Hayes and C. Menzel. IKL specification document. Technical report, IKRIS Interoperability Group, 2006. Unpublished IKRIS memorandum, `http://www.ihmc.us/users/phayes/IKL/SPEC/SPEC.html`.

[10] H. Herre. The Ontology of Mereological Systems. In R. Poli, J. Seibt, M. Healy, and A. Kameas, editors, *Theory and Applications of Ontology – Volume 1: Philosophical Perspectives*. Springer, 2010.

[11] M. Horridge and P. F. Patel-Schneider. OWL 2 web ontology language: Manchester syntax. Working group note, W3C, October 2009.

[12] I. Horrocks, O. Kutz, and U. Sattler. The Even More Irresistible $\mathcal{SROIQ}$. In *10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR)*, pages 57–67. AAAI Press, June 2006.

[13] O. Kutz, T. Mossakowski, C. Galinski, and C. Lange. Towards a standard for heterogeneous ontology integration and interoperability. In K.-S. Choi et al., editors, *First International Conference on Terminology, Language and Content Resources (LaRC)*, pages 101–110. KATS, June 2011.

[14] O. Kutz, T. Mossakowski, and D. Lücke. Carnap, Goguen, and the Hyperontologies: Logical Pluralism and Heterogeneous Structuring in Ontology Design. *Logica Universalis*, 4(2):255–333, 2010. Special Issue on 'Is Logic Universal?'.

[15] O. Kutz, I. Normann, T. Mossakowski, and D. Walther. Chinese Whispers and Connected Alignments. In *5th International Workshop on Ontology Matching (OM)*, 9th International Semantic Web Conference (ISWC), 2010.

[16] C. Lutz, D. Walther, and F. Wolter. Conservative Extensions in Expressive Description Logics. In *Proceedings of IJCAI-07*, pages 453–458. AAAI Press, 2007.

[17] C. Lutz and F. Wolter. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. *Journal of Symbolic Computation*, 45(2):194–228, 2010.

[18] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. WonderWeb Deliverable D18: Ontology Library. Technical report, ISTC-CNR, 2003.

[19] C. Menzel. Metatheory and Motivations for ISO Common Logic. Unpublished draft, 2012.

[20] T. Mossakowski and O. Kutz. The Onto-Logical Translation Graph. In *Modular Ontologies—5th International Workshop (WoMO)*, volume 230 of *Frontiers in Artificial Intelligence and Applications*, pages 94–109. IOS Press, 2011.

[21] T. Mossakowski and A. Tarlecki. Heterogeneous logical environments for distributed specifications. In A. Corradini and U. Montanari, editors, *WADT 2008*, number 5486 in Lecture Notes in Computer Science, pages 266–289. Springer, 2009.

[22] B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz. OWL 2 Web Ontology Language: Profiles. Recommendation, W3C, October 2009.

[23] B. Motik, P. F. Patel-Schneider, and B. Cuenca Grau. OWL 2 Web Ontology Language: Direct Semantics. Recommendation, W3C, October 2009.

[24] S. Schulz, B. Suntisrivaraporn, F. Baader, and M. Boeker. SNOMED reaching its adolescence: Ontologists' and logicians' health check. *Int. J. of Medical Informatics*, 78(Supplement 1):S86–S94, 2009.

[25] B. ten Cate, W. Conradie, M. Marx, and Y. Venema. Definitorially Complete Description Logics. In P. Doherty, J. Mylopoulos, and C. Welty, editors, *Proc. of KR 2006*, pages 79–89. AAAI Press, 2006.