DOLCE in OWL: The Core Theory

Daniele Porello[1,2,*], Laure Vieu[3,2], Walter Terkaj[4], Stefano Borgo[2], Francesco Compagno[2] and Emilio M. Sanfilippo[2]

[1]University of Genoa
[2] CNR-ISTC, Laboratory for Applied Ontology
[3]IRIT, CNRS, University of Toulouse
[4]CNR-STIIMA

Abstract

Foundational ontologies, like DOLCE (an ISO standard), are often implemented in applications via their formalisations in the OWL language. These OWL formalisations are approximations of the theory to cope with the limited expressivity of OWL. With this paper, we start presenting a core OWL2 module of DOLCE, and discuss how to extend it in different directions.

After discussing earlier versions of DOLCE in first-order logic and in OWL, we present the architecture of our modular approach. We select a core fragment of DOLCE in OWL2 (termed "DOLCEbasic$_{\mathrm{OWL}}$"), which provides the main taxonomy and binary relations of the foundational ontology. Then, we discuss how to extend DOLCEbasic$_{\mathrm{OWL}}$ with a module for expressing the n-ary relations of DOLCE ($n > 2$). After this, we give a proof that this OWL2 version is compatible with the original version of DOLCE. Finally, we illustrate the functioning of our OWL2 rendering by means of an example. We conclude by discussing a number of other modules to cope with other core concepts and specific domains.

Keywords

Descriptive Ontology for Linguistic and Cognitive Engineering, OWL version, DOLCEbasic$_{\mathrm{OWL}}$, OWL n-ary relation, Ontology modularisation

1. Introduction

Most foundational ontologies have been formalized in rich logical languages, such as quantified modal logic. While an expressive logical language enables the faithful expression of proponents' ontological views and facilitates subtle analysis, the complexity of reasoning tasks often renders the practical application of such services infeasible. For this reason, computationally manageable versions of foundational ontologies have been proposed, in particular as fragments of first-order logic, i.e., Description Logics, and particularly in OWL2.

This paper focuses on the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [1, 2]. DOLCE originated as a formal theory in first-order modal logic, specifically, in the logic QS5. This version is commonly referred to as the 'D18' version, after the technical report (deliverable) where it was first presented [1]. Due to the richness of DOLCE, proving consistency and exhibiting a model of the theory is not easy. A modular proof of consistency was detailed

FOUST VIII: 8th Workshop on Foundational Ontology, held at JOWO 2024: Episode X The Tukker Zomer of Ontology, July 15-19, 2024, Enschede, Netherlands
*Daniele Porello
daniele.porello@unige.it (D. Porello); walter.terkaj@cnr.it (W. Terkaj)
0000-0001-8902-361X (W. Terkaj)
© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

in [3].[1] To achieve that, a number of simplifications of the D18 version have been made, for instance the modal logic part and the axiom schemata of DOLCE have been removed.[2] More recently, a novel version of DOLCE has been designed in Common Logic (also an ISO standard, ISO/IEC 24707) as required for inclusion in the Top-level ontologies standard ISO 21838.[3] This novel version, termed "DOLCE Simple" is available online.[4]

The format of this version includes the rendering of DOLCE Simple in Common Logic (CLIF[5]). The theory does not use the additional features of Common Logic that extend first-order logic, it is indeed a first-order theory. Moreover, this version have been presented, beside in CLIF, in the usual input formats for theorem prover (tptp, PROVER9). Therefore, throughout this paper we denote this theory by $\text{DOLCEsimple}_{\text{FOL}}$.

$\text{DOLCEsimple}_{\text{FOL}}$ is based on Kutz and Mossakowski's version [3], although a few further simplifications w.r.t D18 were required and a few axioms have been added to enhance the proximity to D18. The objective of $\text{DOLCEsimple}_{\text{FOL}}$ is to enable standard model finders (such as MACE4) to be capable of returning at least a "small model" of DOLCE.[6] Other foundational ontologies follow suit to facilitate the application of provers and model finders, such as BFO [4], TUPPER [5], and UFO [6]. For an overview and test of these implementations, see [7].

A requirement of this OWL2 version is the 'compatibility' with the (first-order logic) version submitted in CLIF. Compatibility here means that the translation into first-order logic of the axioms of the OWL version of DOLCE can be proved from the (first-order) CLIF version $\text{DOLCEsimple}_{\text{FOL}}$. That is, every model of $\text{DOLCEsimple}_{\text{FOL}}$ is also a model of the OWL version. Although the OWL version is less specific, as expected, and allows for many more, possibly unintended, models, the models of the original ontology are preserved. This result was achieved for $\text{DOLCEsimple}_{\text{FOL}}$ and a preliminary OWL2 version of DOLCE: the proof that the CLIF version entails the OWL2 version was done by means of PROVER9 (after translating both CLIF and OWL into first-order logic)[7].

A number of OWL versions of DOLCE, inspired by D18, have been developed in the past. Notably, the first OWL version of DOLCE, i.e. the DOLCE lite version [8]; see [2] for a chronology of previous OWL versions of DOLCE. A preliminary OWL2 version of DOLCE Simple was also developed for the inclusion in ISO 21838 standard[8].

In this paper, starting from the OWL2 version developed for the ISO 21838, we document our advancement in the project of developing an OWL2 version of DOLCE. The main motivation for the present novel proposal is driven by the intention to present DOLCE in OWL2 as a basic theory (as faithful as possible to the original DOLCE in the D18 and justified by the axioms of D18), that can be expanded for practical applications via a library of DOLCE modules, either subtheories or

[1] For an implementation of this strategy, see also https://github.com/spechub/Hets-lib/tree/master/Ontology/Dolce
[2] A comprehensive documentation of the simplifications of this version w.r.t the D18 is available at https://github.com/spechub/Hets-lib/blob/master/Ontology/Dolce/DolceSimpl.dol
[3] https://www.iso.org/standard/78927.html
[4] http://www.loa.istc.cnr.it/index.php/dolce/. See also the repository COLORE, https://github.com/gruninger/colore/tree/master/ontologies/dolce. This version has been expanded to discuss the alignments between foundational ontologies in the OntoCommons project, cf. https://zenodo.org/records/10894153.
[5] https://www.iso.org/standard/66249.html
[6] For the full list of simplifications w.r.t D18 see the documentation at http://www.loa.istc.cnr.it/index.php/dolce.
[7] See footnote 3.
[8] See http://www.loa.istc.cnr.it/index.php/dolce.

extensions, also in OWL2. For this reason, we will explore the conceptual expressivity of OWL2 to capture, as far as possible, the original spirit of the ontological analysis presented in the D18.

In this paper, we will introduce and discuss a core theory, here termed "DOLCEbasic$_{\text{OWL}}$", which includes the main taxonomy of DOLCE and axiomatises, as far as possible, the main classes and a rich number of binary relations of DOLCE. The use of this theory is illustrated on an example taken from the literature. We also discuss the modular approach and the strategy adopted to extend this core module. In particular, a module to handle the n-ary relations ($n > 2$) of DOLCE, among which the temporalised relations (e.g. temporary parthood, constitution, paticipation, etc.) is planned. These implementations are available online.[9]

The paper is organised as follows. Section 2 presents the overall approach to develop the OWL version of DOLCE, and Section 3 illustrates the formalization in OWL2. The proposed ontology is tested by automatically proving its axioms from the axioms of DOLCEsimple$_{\text{FOL}}$ in Section 4, and is exemplified via a use case in Section 5. Finally, Section 6 draws the conclusions.

2. Approach

Our goal is to develop an OWL2 version of DOLCE that is close to and justified by the original theory presented in D18 [1] and simplified in DOLCEsimple$_{\text{FOL}}$.Specifically, we are working with a fragment of the Description Logic $\mathcal{SROIQ}$, a sub-logic of OWL2[9][10].

It is well-established that the expressive power of $\mathcal{SROIQ}$ does not allow for directly writing the first-order axioms (and theories) required by the D18 version of DOLCE. This is caused by two types of restrictions, which are imposed to guarantee the decidability of $\mathcal{SROIQ}$ ontologies: a *syntactic* restriction on single formulas, which does not allow for writing, e.g., n-ary predicates, for $n \geq 3$ and a structural restriction on theories, specifically on the *role box*, which demands *regularity* and *simplicity*, cf. [9]. Regularity demands only simple dependencies between roles in the role hierarchy and is a structural property of the role box.

Methodologies for approximating FOL-theories with $\mathcal{SROIQ}$ have been developed for instance in [10] and [11]. The strategy in [10] involves checking a large number of candidates $\mathcal{SROIQ}$ theories and assess which of them are "closer" to the original FOL-theory by experimenting on automatically generated models.

Here, our aim is not to find the "closest" $\mathcal{SROIQ}$ version to the FOL-DOLCE. Instead, we will select the axioms or theorems from the D18 version that we wish to keep in OWL2, focusing on the practical use and understandability of such constraints and axioms.

An obvious constraint when approximating a FOL theory in OWL2 is that all description logic languages are limited to binary relations, so n-ary relations ($n > 2$) cannot be directly represented as such. DOLCE extensively uses temporalised relations as a major effort has been dedicated to axiomatise how endurants and perdurants behave across time. Those temporalised relations are all at least ternary relations. In addition, mereological operators (e.g., the mereological sum) are at least ternary relations. In the modular approach chosen, we firstly develop the basic core module (DOLCEbasic$_{\text{OWL}}$) presented in this paper, which is limited

[9] https://github.com/appliedontolab/DOLCE.

[10] Protégé classifies the Description Logic Expressivity of DOLCEbasic$_{\text{OWL}}$ as the logic $\mathcal{SRIF}$. In fact, we are using transitive roles (aka object properties), complex roles inclusions, inverse roles, functionality of roles.

to: i) the original binary relations of DOLCE, plus the ii) the *constant* versions of the temporalised relations of DOLCE. For instance, the *participation* of an endurant to a perdurant at a time in DOLCE (PC, in D18), appears in DOLCEbasic$_{\mathrm{OWL}}$ as the object property (i.e. binary) `constantParticipantOf`. Another module, termed DOLCE N-ARY REL, not presented in this paper, will be dedicated to properly introduce and treat the n-ary relations (for $n > 2$) in OWL2. We shall follow the "reification" approach advocated by the W3C[11], inspired by the neo-Davidsonian approach to handling events and their arguments in natural language semantics. This implies the introduction of a new category of entities called "relation instances" (not included in DOLCEbasic$_{\mathrm{OWL}}$), to which each argument of the original relation will be related, by a distinct new binary relation. For example, the mereological sum[12] between perdurants or abstracts is defined by a ternary relation $\mathsf{sum}(x, y, z)$ (indicating that the sum of x and y results in z). Mereological sums will be represented in DOLCE N-ARY REL by the following assertions.

$$\mathsf{SumInstance}(r), \mathsf{sumAddendum1}(r, x), \mathsf{sumAddendum2}(r, y), \mathsf{sumResult}(r, z)$$

The class SumInstance will be then a subclass of the relation instances. An important ontological question regards the nature of these relation instances: whether this new category fits within the original DOLCE taxonomy, and if so, under which main category, and if not, how to handle it. An option is to view a relation instance as some sort of "state of the relation holding" aka "situation"[13] or "state of affairs", or "facts". Hence, the question is whether, in DOLCE terms, they can be seen as perdurants or as abstracts (states and situations, which are in time, are perdurants in DOLCE which are disjoint with facts, which are out of time and thus abstracts in DOLCE). Considering them as perdurants is in line with the neo-Davidsonian approach to events, and seems quite natural for temporalised relations and perhaps for ternary relations between perdurants. However, it is not general enough, as it appears inappropriate for ternary relations between abstracts such as the mereological sum over spatial regions, which are intended to be out of time in DOLCE. Furthermore, a delicate aspect arises when considering the axiom of DOLCE which states that all perdurants have endurants that participate in them at some time. If we consider relation instances as perdurants, they too must adhere to that axiom, potentially leading to a regress. On the other hand, considering relation instances as abstracts again forces us to question the DOLCE axioms for mereology, on pain of a potential regression: if r identifies the sum of a and b, then we have to introduce an identifier s for the sum of r and a, then one for the sum of s and r, and so on.

We thus handle relation instances as mere technical additions to the ontology, unrelated with the original rationale of the taxonomy. One is forced to introduce them in OWL versions of DOLCE because of the limited expressivity of $\mathcal{SROIQ}$, not for genuine ontological reasons.

Several other approaches, with and without reification, to deal with temporalised relations in OWL have been discussed [12]. In particular, the "Temporally Qualified Continuants Pattern" approach considers that the temporal argument can be embedded in the other arguments by considering their relevant "phases". Handling ternary temporalised relations directly as binary ones in OWL has the considerable advantage of enabling the expression of transitivity and other

[11]https://www.w3.org/TR/swbp-n-aryRelations
[12]The binary operator $+$ in the D18 can be equivalently encoded as a ternary relation, as usual.
[13]See, e.g., https://nemo-ufes.github.io/gufo/#situations

properties supporting reasoning, which is extremely limited with the standard W3C approach. Unfortunately, the Temporally Qualified Continuants Pattern approach cannot be generalised in OWL to non-temporalised ternary relations, such as the mereological sum between abstracts, nor to temporalised relations of arity above 3, such as the temporalised mereological sum between endurants. The adopted modular approach will allow for adding still other modules to extend DOLCEbasic$_{\mathrm{OWL}}$ with core concepts or domain-level concepts.

3. OWL2 Formalization

We now turn to illustrate and motivate the main features of the core module DOLCEbasic$_{\mathrm{OWL}}$.

3.1. Class and Property Hierarchy

The taxonomy of the proposed DOLCEbasic$_{\mathrm{OWL}}$ ontology is represented in Figure 1. There are a few differences with respect to the DOLCE taxonomy shown in D18: `Atom` and `ConstantAtom` are not new categories, as they can be defined in DOLCE and are not disjoint with their siblings.[14]

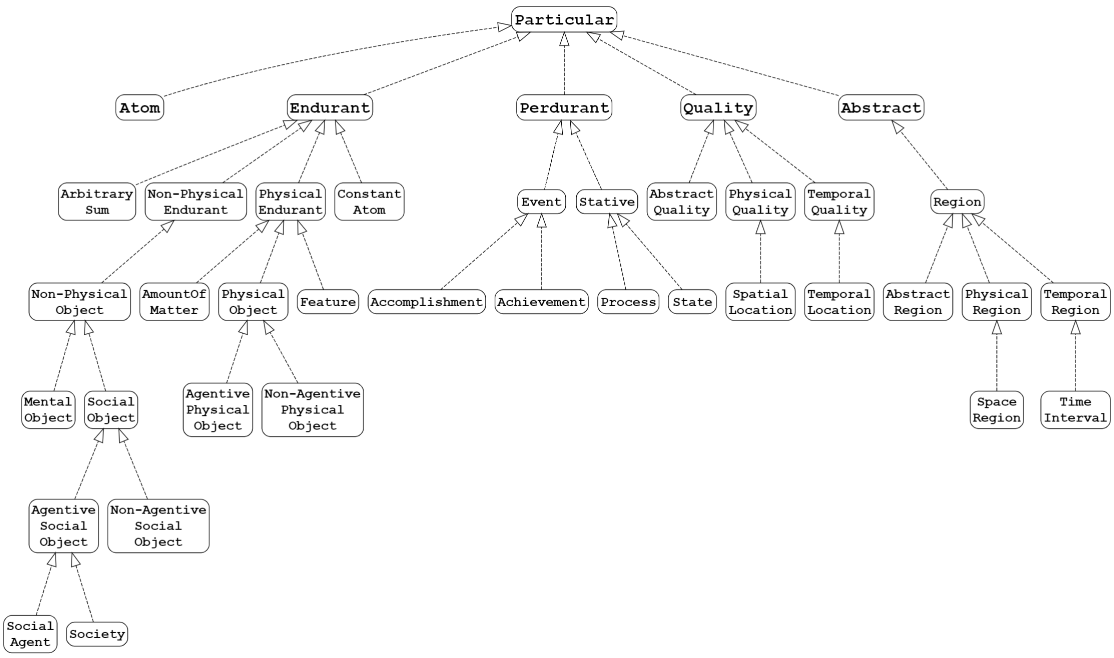

Figure 1: DOLCEbasic$_{\mathrm{OWL}}$ class hierarchy. Rounded boxes represent classes identified by their `rdfs:label`. Dashed arrows represent `rdfs:subClassOf` relations.

The hierarchy of the object properties in DOLCEbasic$_{\mathrm{OWL}}$ is represented in Figure 2. Notice the constant versions of the temporalised relations of DOLCE, some of them not defined in D18, like `constantlyOverlaps` or `constantAtomicPartOf`, although they are definable in DOLCE.

[14] A definition of this categories is included in DOLCEsimple$_{\mathrm{FOL}}$, see https://github.com/appliedontolab/DOLCE/tree/main/OWL/Proof.

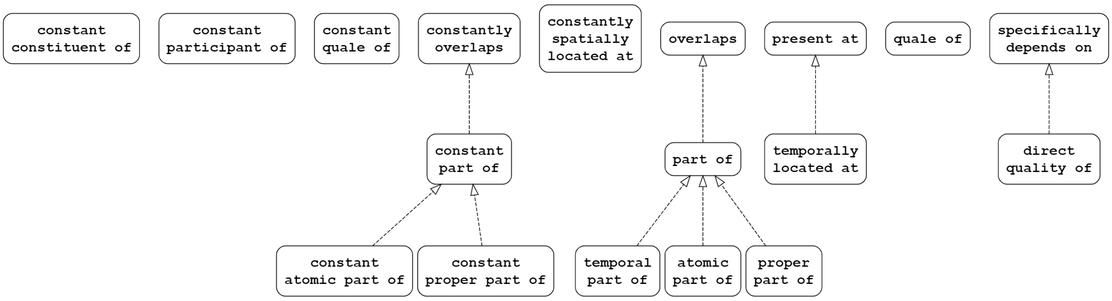

Figure 2: DOLCEbasic$_{\mathrm{OWL}}$ property hierarchy. Rounded boxes represent object properties identified by their `rdfs:label`. Dashed arrows represent `rdfs:subPropertyOf` relations.

3.2. Focus on Endurants and Perdurants

Figure 3 shows the possible relations for classes `Endurant` and `Physical Endurant` in DOLCEbasic$_{\mathrm{OWL}}$, such as *parthood* (`constantPartOf`, `constantProperPartOf`), *constitution* (`constantConstituentOf`), *presence* (`presentAt`, `temporallyLocatedAt`), *participation* (`constantParticipationOf`), *spatial location* (`constantlySpatiallyLocatedAt`), and *quality* characterization (`directQualityOf`).

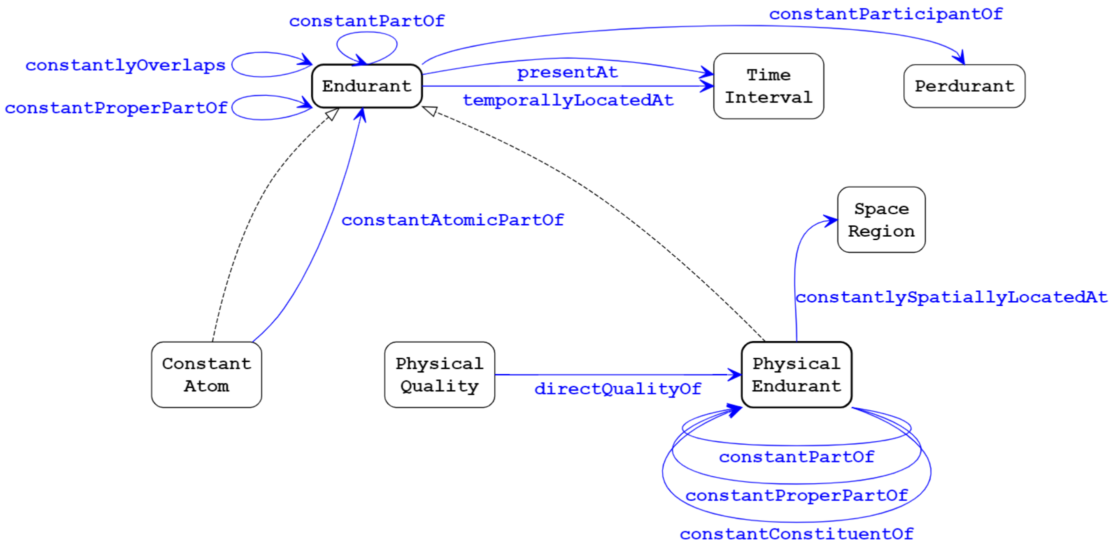

Figure 3: Relations for `Endurant` and `PhysicalEndurant` in DOLCEbasic$_{\mathrm{OWL}}$. Rounded boxes represent classes identified by their `rdfs:label`. Blue solid arrows represent possible relations labelled with the corresponding object property, while dashed arrows represent `rdfs:subClassOf` relations.

Figure 4 illustrates the possible relations for class `Perdurant`, including *parthood* (`partOf`, `properPartOf`,`temporalPartOf`), *presence* (`presentAt`, `temporallyLocatedAt`), *participation* (`constantParticipationOf`), *spatial location* (`constantlySpatiallyLocatedAt`), *quality* characterization (`directQualityOf`).

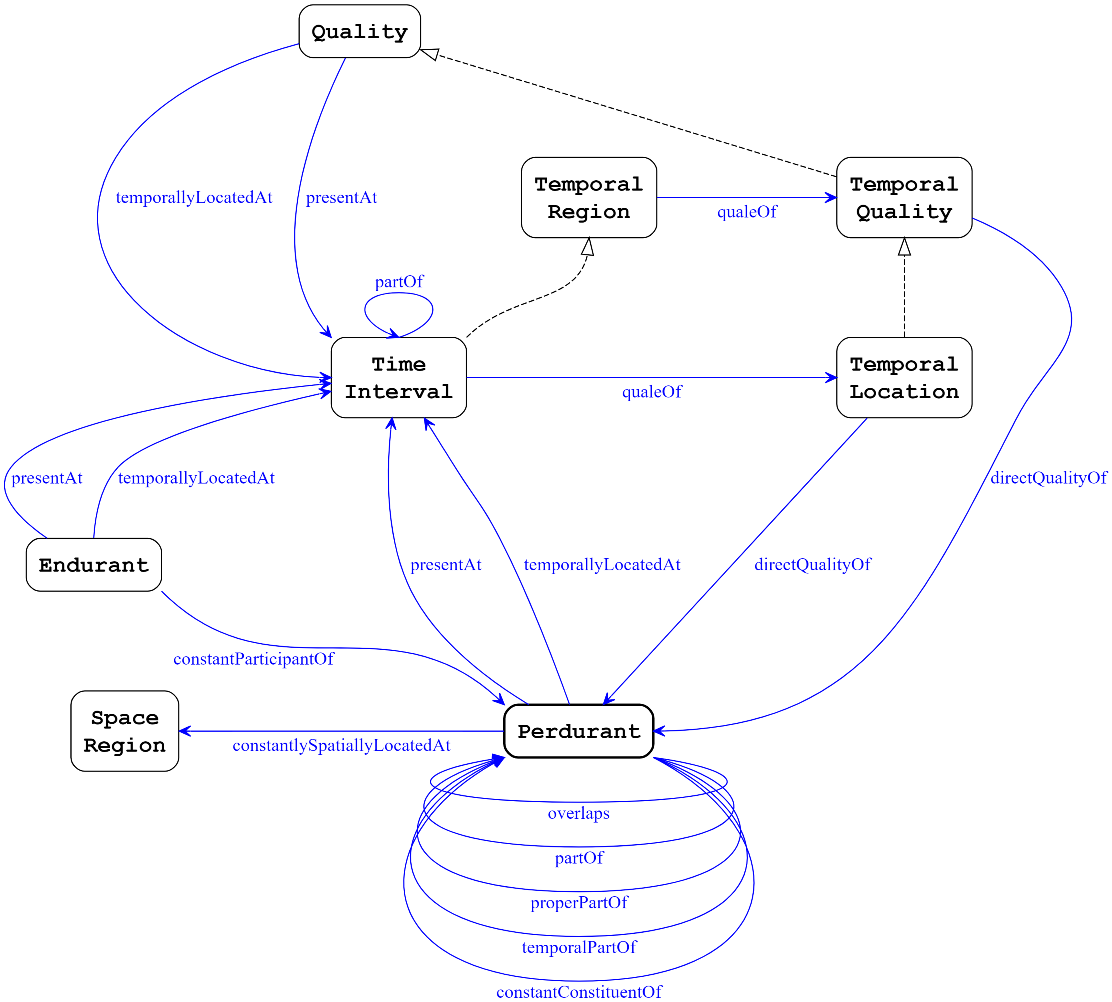

Figure 4: Relations for `Perdurant` in DOLCEbasic$_{\text{OWL}}$. Rounded boxes represent classes identified by their `rdfs:label`. Blue solid arrows represent possible relations labelled with the corresponding object property, while dashed arrows represent `rdfs:subClassOf` relations.

3.3. Focus on Time

DOLCE provides a rich representation of how entities behave through time, and this attention to time is present in this OWL2 rendering although in DOLCEbasic$_{\text{OWL}}$ we only have binary constant versions. Here we intend to highlight how some of those temporal aspects are encoded on DOLCEbasic$_{\text{OWL}}$, what could be implemented, and what could not.

As sketched in Figure 4, a perdurant has temporal qualities (e.g. the duration of an event), among which a unique quality in the class `TemporalLocation`. The temporal location of a perdurant is related, through `qualeOf`, to the unique time interval which is the whole temporal extension of the perdurant. Time intervals are abstract entities in DOLCE, they are mereologically organised by the binary `partOf` relation. They can be atomic and non-convex, i.e., scattered through time (as generalised intervals). The perdurant is also directly related to the same time interval of its whole temporal extension through `temporallyLocatedAt`. Also endurants and qualities are temporally located at a time interval, their temporal extension.[15] The relation `presentAt` relates perdurants, endurants and qualities to any time interval that is part of their whole temporal extension.

Although the expressivity of OWL2 doesn't enable a proper characterization of mereological relations[16] and despite the simplicity and the regularity constraints on properties in OWL2, we were able to guarantee a few facts.

Here, we opt to keep the transitivity of `partOf` and `properPartOf` and to drop the irreflexivity of `properPartOf`. We have also to drop the reflexivity of `partOf` because the reflexivity is implemented in Protégé *globally*, while `partOf` doesn't range over the whole of `owl:Thing`, but only over perdurants and abstracts. To implement *local* reflexivity, with the use of the $\exists R.Self$ $\mathcal{SROIQ}$ constructs, requires a simple property R, which is unfortunately incompatible with transitivity. Antisymmetry is impossible to express. Moroever, `partOf` is a sub-property of `overlaps`, which is symmetric, but we can only partially enforce the characterisation of overlaps in terms of parthood. Similar observations hold regarding constant versions of temporalised mereological relations over endurants such as `constantPartOf`.

Moreover, that the temporal extension is the maximal time interval at which an entity is present is somehow captured, not directly by an axiom on `presentAt`, `temporallyLocatedAt` and `partOf` (enforcing the dissectivity of `presentAt` would cause irregularity), but by a similar axiom on `overlaps`. We can also enforce that a quality and its bearer share the same temporal extension. On the other hand, these expressivity limitations do not allow for enforcing the unicity of the temporal extensions, nor acceptable characterisations of other classes and object properties. For instance, `temporalPartOf` simply is a subproperty of `partOf` between perdurants, failing to grasp the fact that a temporal part of a perdurant is a "temporal slice" of it, i.e., a maximal part during some time interval.

Several constant versions of the temporalised relations defined in DOLCE, with a specific temporal pattern regarding the presence of their arguments, have been introduced. For instance, `constantPartOf` and `constantConstituentOf` are defined such that the part is present when the whole is present, and the substrate is present when the entity it constitutes is present (see section 4). When completing with constant versions of all temporalised relations, a decision has to be made on which argument has its temporal extension included in the other's or what other pattern is suitable. For `constantlyOverlaps`, which is symmetric, the decision has to be compatible with `constantPartOf` which is a sub-property with its own temporal pattern. The option that both arguments of `constantlyOverlaps` share the same temporal extension is rejected as it forces to change the definition of `constantPartOf`. As seen in next section, an axiom enforcing that the temporal extensions of both arguments overlap is expressible in OWL2, but not supported by the reasoner Pellet in Protégé. We are therefore left with the characterization of `constantlyOverlaps` in terms of the arguments sharing a constant part, without being able to exclude a model in which the two arguments are temporally disjoint.

---

[15]The time intervals at which endurants, perdurants and qualities are temporally located are sometimes called the "temporal locations" of such entities. In order not to confuse them with the temporal qualities belonging to the `TemporalLocation` class, we will here stick to the term "temporal extension".

[16]This fact is well-known, see [13].

4. Proof

To establish the adequacy of our OWL version of DOLCE, we prove that the first-order logic version DOLCEsimple$_{\text{FOL}}$ entails the axioms of DOLCEbasic$_{\text{OWL}}$. To achieve that, our strategy is to translate DOLCEbasic$_{\text{OWL}}$ into first-order logic, then use Prover9[17] to automatically prove each axiom of DOLCEbasic$_{\text{OWL}}$ from DOLCEsimple$_{\text{FOL}}$. The translation from DOLCEbasic$_{\text{OWL}}$ into first-order logic was done automatically by means of the translation tool developed in [14].[18] A minor technicality is that DOLCEsimple$_{\text{FOL}}$ uses the original abbreviated D18 labels of classes and relations (e.g., `ed` for `Endurant`, `p` for `partOf`), while DOLCEbasic$_{\text{OWL}}$ employs fully spelled-out labels to comply with standard use. Thus, to perform the proof, we translated the alphabet of DOLCEbasic$_{\text{OWL}}$ into the alphabet of DOLCEsimple$_{\text{FOL}}$.

Prover9 is able to directly prove each axiom of DOLCEbasic$_{\text{OWL}}$ that does not involve the constant version of the temporalised relations of DOLCE (e.g. constantParticipantOf), which are novel to the DOLCEbasic$_{\text{OWL}}$, as expected. To prove the axioms of DOLCEbasic$_{\text{OWL}}$ involving constant temporalised relations, we added their definitions to the theory of DOLCEsimple$_{\text{FOL}}$. Notice that such relations are definable in DOLCE, since they always use relations and classes that have been already axiomatised by the theory (e.g. mereology, participation, or constitution). For instance, constant part of and constantly overlaps are captured by means of definitions of the following form (omitting outermost universal quantifications).

$$\mathtt{constantPartof}(x,y) \leftrightarrow (\mathtt{ed}(x) \land \mathtt{ed}(y) \land \exists t \mathtt{pre}(y,t) \land \forall t(\mathtt{pre}(y,t) \rightarrow \mathtt{tp}(x,y,t)))$$

$$\mathtt{constantlyOverlaps}(x,y) \leftrightarrow (\mathtt{ed}(x) \land \mathtt{ed}(y) \land \exists t(\mathtt{pre}(x,t) \land \mathtt{pre}(y,t)) \land \exists z(\mathtt{cp}(z,x) \land \mathtt{cp}(z,y))$$

where `ed`, `pre`, `tp`, `tpp` are the class `Endurant`, the relation `presentAt`, the relations of temporary parthood and temporary proper parthood defined in DOLCEsimple$_{\text{FOL}}$ (respectively), and `cp` stands for `constantPartOf`.[19]

The definitions for constant atom, constant atomic part, and constant proper part follow the pattern of `constantPartOf`; those of `constantlyOverlaps` is motivated by the goal of proving symmetry and compatibility with `constantPartOf`. Notice that, in DOLCEsimple$_{\text{FOL}}$, we can also prove that the temporal extension of two constantly overlapping entities overlaps, however this fact cannot be enforced in OWL for the reasons at the end of Section 3.3.[20]

The constant versions of participation, constitution, quale of, and spatial location follow a similar pattern. We present the definition of `constantConstituentOf` and we refer to the documentation for the other definitions.

$$\mathtt{constantConstituentOf}(x,y) \leftrightarrow (\mathtt{ped}(x) \land \mathtt{ped}(y)) \lor (\mathtt{nped}(x) \land \mathtt{nped}(y)) \lor (\mathtt{pd}(x) \land \mathtt{pd}(y)) \land \exists t \mathtt{pre}(y,t) \land \forall t(\mathtt{pre}(y,t) \rightarrow \mathtt{k}(x,y,t))$$

[17] https://www.cs.unm.edu/~mccune/prover9/

[18] The Python package is available at https://github.com/gavel-tool/python-gavel-owl/blob/dev/README.rst

[19] Notice that the relation of constant parthood was already present in D18, cf. Axiom Dd25.

[20] Formula $\forall t \forall t'(\exists x \exists y(\mathtt{temporallyLocatedAt}(x,t) \land \mathtt{cov}(x,y) \land \mathtt{temporallyLocatedAt}(y,t')) \rightarrow \mathtt{ov}(t,t')))$ is a theorem of DOLCEsimple$_{\text{FOL}}$ and could be added to DOLCEbasic$_{\text{OWL}}$, but it clashes with Pellet in Protégé (ver. 5.6.3), so we omit it.

Here, `ped` corresponds to `PhysicalEndurant`, `nped` to `NonPhysicalEndurant`, `pd` to `Perdurant`, `k` is (time-dependent) constitution, defined in DOLCEsimple$_{\text{FOL}}$.

The proofs, the documentation, and all the required files are available online[21]. In particular, the repository contains: i) the version of DOLCEsimple$_{\text{FOL}}$ in the format of PROVER9/MACE4 and the proof of its consistency; ii) the translation of DOLCEbasic$_{\text{OWL}}$ into first-order logic, in the format of PROVER9, iii) a report of the axioms of DOLCEbasic$_{\text{OWL}}$ proved from DOLCEsimple$_{\text{FOL}}$.

5. Use Case

This section outlines a use case focusing on composition and constitution, illustrating how DOLCEbasic$_{\text{OWL}}$ can be instantiated to demonstrate its effectiveness. The reference case, derived from Case n.1 in Borgo et al. [2], deals with the modeling of a physical object like a table. The table and its components are artefacts, i.e., intentionally produced products. For the sake of the example, it is assumed that a table is identified through time by its tabletop; also, the quantity of wood constituting the table changes if a leg is substituted. Accordingly, the existence of the table does not imply that it is made of the same matter throughout its whole life. The table undergoes three lifecycle phases:

P1 Construction of a wooden table (`T`) consisting of a table top (`Ttop`) and four legs (`Leg1, Leg2, Leg3, Leg4`) during time interval `t0`. The tabletop and legs are made of wood material (`Wtop, W1, W2, W3, W4`, respectively).

P2 One table leg (i.e. `Leg4`) is replaced by a new leg (i.e. `Leg4new` made of wood material `W4new`) during time interval `t1`.

P3 The table is dismantled, resulting in its cessation of existence, while the wood remains intact during time interval `t2`.

Herein, we focus only on phase *P1*, making the assumption that the table will not change its composition in the future. This simplified use case is instantiated by module *Case01basic*[22] importing DOLCEbasic$_{\text{OWL}}$, thus representing only constant binary relations between instances. The instantiation of the use case can be further extended in the future taking full advantage of the modular approach presented in Sect.2 to represent the evolution of the table along all phases *P1*-*P3*; indeed, the ternary relations from DOLCE N-ARY REL are needed to represent time-based relations. Table 1 lists the relevant namespace prefix bindings. All modules are available online.

The instances representing the use case belong to the following classes:

- `TimeInterval` from DOLCEbasic$_{\text{OWL}}$.
- `Artefact` class, defined in module *Case01basic* according to Borgo and Vieu [15], as a subclass of `PhysicalObject`.
- Subclasses of `Artefact` (i.e., `Table`, `TableTop`, and `TableLeg`) and `AmountOfMatter` (i.e. `Wood`) specifically defined in module *Case01basic* for this use case.

[21] https://github.com/appliedontolab/DOLCE/tree/main/OWL/Proof
[22] https://w3id.org/DOLCE/OWL/UC/Case01basic

Table 1
List of namespaces with prefix names

Prefix	**IRI**
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs:	http://www.w3.org/2000/01/rdf-schema#
owl:	http://www.w3.org/2002/07/owl#
xsd:	http://www.w3.org/2001/XMLSchema#
db:	https://w3id.org/DOLCE/OWL/DOLCEbasic#
c01:	https://w3id.org/DOLCE/OWL/UC/Case01basic#

Figure 5 shows the novel classes specializing DOLCEbasic$_{\text{OWL}}$ and the related instances that are defined in module *Case01basic*.

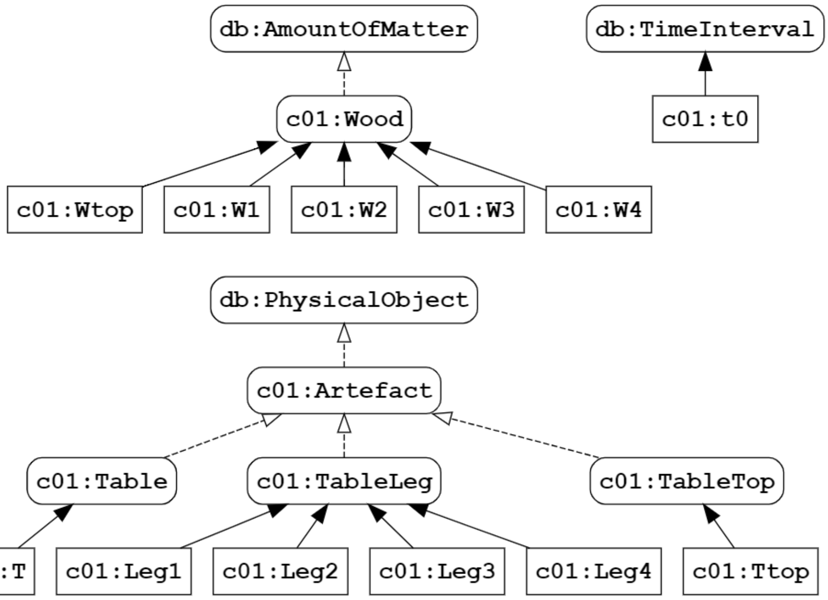

Figure 5: Graphical representation of *Case01basic*. Rounded boxes stand for classes, boxes for instances, dashed arrows for `rdfs:subClassOf` relations, and solid arrows for `rdf:type` relations.

Three types of relations are needed to complete the instantiation of the use case:

- *Presence* of artefacts and amount of matter during a time interval.
- *Parthood*, applying to entities of the same category. This relation is needed to formally represent the components of the table.
- *Constitution*, which is needed to specify the material of the table components.

Presence is a binary relation defined in DOLCEbasic$_{\text{OWL}}$ (cf. `presentAt` in Figure 3). Parthood and constitution can be modeled as either constant or time-dependant relations. In the first case, a binary relation suffices and the related OWL2 object properties are introduced in DOLCEbasic$_{\text{OWL}}$ (cf. `constantProperPartOf` and `constantConstituentOf` in Figure 3).

Focusing on a specific leg of the table (i.e., `Leg1`), we can show how the relations between instances can be represented taking advantage of DOLCEbasic$_{\mathrm{OWL}}$ (see Figure 3). A similar pattern applies to the other table components (i.e., `c01:Ttop`, `c01:Leg2`, `c01:Leg3`, `c01:Leg4`).

The use case instantiation of module *Case01basic* has been tested with the reasoner Pellet available for Protégé and no inconsistency was detected.

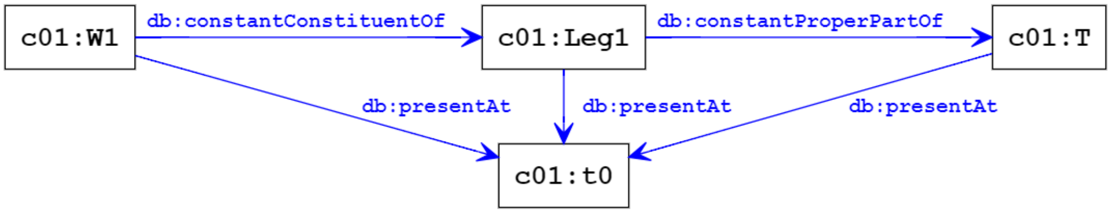

Figure 6: Graphical representation of *Case01basic* relations. Boxes stands for instances and blue solid arrows for relations labelled with the corresponding object property.

6. Conclusions and further work

We presented the core aspects of the OWL2 version of the DOLCE ontology, in particular, the DOLCEbasic$_{\mathrm{OWL}}$ module. This provides the taxonomy of DOLCE, its main binary relations, and a formal representation of the axioms of DOLCE in the expressivity of OWL2. Differently from other existing Semantic Web formal representations of DOLCE, DOLCEbasic$_{\mathrm{OWL}}$ is a closer match with the original D18 version of the ontology; for instance, differently from other versions like the one presented in [8], it does not include modeling elements that are not present in the D18.

As said, DOLCEbasic$_{\mathrm{OWL}}$ is the first 'basic' module in a library of modules for the exploitation of DOLCE through the Semantic Web. The library will (at least) include the DOLCE N-ARY REL module for the representation of predicates with arity higher than 2, for instance, to include time parameters. This will allow for the representation of phenomena concerning objects' change through time, e.g., change in parts or material constitution, which play a fundamental role in multiple application contexts. In compliance with DOLCE-driven research, another module will comprise the representation of *concepts* and *descriptions* [16], which often find their place for representing (social) roles but also engineering technical specifications [17], among other entities. A further module will introduce the use of OWL2 data properties, especially for enriching the representation of qualities and qualia when the latter are characterized through numerical values. Last but not least, current research aims at developing modules for specific application domains, including product design and manufacturing [17].

From an ontology design perspective, the development of the library of modules will take the advantage of existing Semantic Web resources like the W3C Time Ontology[23] and SSN/SOSA,[24] which we will attempt to (at least partially) integrate into the modules.

[23]https://www.w3.org/TR/owl-time/.
[24]https://www.w3.org/TR/vocab-ssn/.

Acknowledgments

We wish to thank colleagues and friends at the CNR-ISTC Laboratory for Applied Ontology, Claudio Masolo, Roberta Ferrario, and Nicola Guarino, for their contributions and discussions.

References

[1] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, WonderWeb Deliverable D18, Technical Report, CNR, 2003.

[2] S. Borgo, R. Ferrario, A. Gangemi, N. Guarino, C. Masolo, D. Porello, E. M. Sanfilippo, L. Vieu, DOLCE: A descriptive ontology for linguistic and cognitive engineering, Applied Ontology 17 (2022) 45–69. doi:10.3233/AO-210259.

[3] O. Kutz, T. Mossakowski, A modular consistency proof for DOLCE, in: W. Burgard, D. Roth (Eds.), Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011, AAAI Press, 2011. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3754.

[4] J. N. Otte, J. Beverley, A. Ruttenberg, Bfo: Basic formal ontology, Applied ontology 17 (2022) 17–43.

[5] M. Grüninger, Y. Ru, J. Thai, Tupper: A top level ontology within standards, Applied Ontology 17 (2022) 143–165.

[6] G. Guizzardi, A. Botti Benevides, C. M. Fonseca, D. Porello, J. P. A. Almeida, T. Prince Sales, Ufo: Unified foundational ontology, Applied ontology 17 (2022) 167–210.

[7] P. Garbacz, Foundational ontologies meet automatic theorem provers, in: T. P. Sales, et al. (Eds.), Proceedings of the Joint Ontology Workshops 2022 Episode VIII: The Svear Sommar of Ontology, Jönköping, Sweden, August 15-19, 2022, volume 3249 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022. URL: https://ceur-ws.org/Vol-3249/paper1-FOUST.pdf.

[8] V. Presutti, A. Gangemi, Dolce+D&S Ultralite and its main Ontology Design Patterns., Ontology Engineering with Ontology Design Patterns 25 (2016) 81–103.

[9] I. Horrocks, O. Kutz, U. Sattler, The even more irresistible SROIQ, in: P. Doherty, J. Mylopoulos, C. A. Welty (Eds.), Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006, AAAI Press, 2006, pp. 57–67. URL: http://www.aaai.org/Library/KR/2006/kr06-009.php.

[10] A. B. Benevides, J. Bourguet, G. Guizzardi, R. Peñaloza, J. P. A. Almeida, Representing a reference foundational ontology of events in SROIQ, Appl. Ontology 14 (2019) 293–334. URL: https://doi.org/10.3233/AO-190214. doi:10.3233/AO-190214.

[11] T. Hahmann, R. W. P. II, Automatically extracting OWL versions of FOL ontologies, in: A. Hotho, E. Blomqvist, S. Dietze, A. Fokoue, Y. Ding, P. M. Barnaghi, A. Haller, M. Dragoni, H. Alani (Eds.), The Semantic Web - ISWC 2021 - 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24-28, 2021, Proceedings, volume 12922 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 252–269. URL: https://doi.org/10.1007/978-3-030-88361-4_15. doi:10.1007/978-3-030-88361-4_15.

[12] P. Garbacz, R. Trypuz, Representation of tensed relations in owl: A survey of philosophically-motivated patterns, in: Metadata and Semantic Research: 11th Inter-

national Conference, MTSR 2017, Tallinn, Estonia, November 28–December 1, 2017, Proceedings 11, Springer, 2017, pp. 62–73.

[13] C. M. Keet, O. Kutz, Orchestrating a network of mereo(topo)logical theories, in: Proceedings of the 9th Knowledge Capture Conference, K-CAP '17, Association for Computing Machinery, New York, NY, USA, 2017. URL: https://doi.org/10.1145/3148011.3148013. doi:10.1145/3148011.3148013.

[14] S. Flügel, A. Kleinau, F. Neuhaus, M. Glauer, J. Hastings, FOWL - an OWL to FOL translator, in: E. M. Sanfilippo, et al. (Eds.), Proceedings of the Joint Ontology Workshops 2021 Episode VII: The Bolzano Summer of Knowledge co-located with the 12th International Conference on Formal Ontology in Information Systems (FOIS 2021), and the 12th International Conference on Biomedical Ontologies (ICBO 2021), Bolzano, Italy, September 11-18, 2021, volume 2969 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021. URL: https://ceur-ws.org/Vol-2969/paper7-DEMO.pdf.

[15] S. Borgo, L. Vieu, Artefacts in Formal Ontology, in: A. Meijers (Ed.), Philosophy of Technology and Engineering Sciences, Handbook of the Philosophy of Science, North-Holland, Amsterdam, 2009, pp. 273–307. doi:10.1016/B978-0-444-51667-1.50015-X.

[16] C. Masolo, L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gangemi, N. Guarino, et al., Social roles and their descriptions., in: KR, 2004, pp. 267–277.

[17] W. Terkaj, S. Borgo, E. Sanfilippo, Ontology for Industrial Engineering: A DOLCE Compliant Approach, in: CEUR Workshop Proceedings, volume 3240, 2022.