Approximating DOLCE in OWL: The **DOLCEbasic** and **DOLCEnaryRel** Core Modules

Daniele Porello[1], Walter Terkaj[2], Laure Vieu[3], Emilio Sanfilippo[4], and Francesco Compagno[5]

[1]DAFIST, Università di Genova, Italy
[2]STIIMA, CNR, Italy
[3]IRIT, CNRS, France
[4]ISTC Laboratory for Applied Ontology, CNR, Italy
[5]Doctorate Program in Industrial Innovation, University of Trento,Italy

March 12, 2026

Abstract

Foundational ontologies are usually developed in powerful logical languages, while they are often implemented in applications via their formalisations in the Web Ontology language (OWL). These OWL formalisations are in fact approximations of the original theories, to cope with the well-known limited expressivity of OWL. In this paper, we propose a novel modular approach to the OWL rendering of the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE). We start presenting two fundamental modules of DOLCE in OWL 2: *i*) a core module of DOLCE (termed "DOLCEbasic$_{\text{OWL}}$"), which provides the main taxonomy and the binary relations of the foundational ontology and *ii*) an extension (termed "DOLCEnaryRel$_{\text{OWL}}$") to deal with the n-ary relations of DOLCE (for $n > 2$). We start presenting two fundamental modules of DOLCE in OWL 2: *i*) a core module of DOLCE (termed "DOLCEbasic$_{\text{OWL}}$"), which provides the main taxonomy and the binary relations of the foundational ontology and *ii*) an extension (termed "DOLCEnaryRel$_{\text{OWL}}$") to deal with the n-ary relations of DOLCE (for $n > 2$). We examine how the OWL rendering requires approaching delicate and truly ontological issues to motivate the choices made to circumvent the limited expressivity. To provide a minimal justification of our approximation, we establish that the OWL 2 versions are compatible with the original version of DOLCE by offering an automated proof that the first-order version of DOLCE entails the translations into first-order logic of the OWL 2 modules. Other adequacy criteria are then discussed. Finally, we illustrate the functioning of our rendering by means of examples. We conclude by discussing a number of other modules to cope with other core concepts and specific domains.

Descriptive Ontology for Linguistic and Cognitive Engineering OWL 2 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 (DL), and particularly in OWL 2[1].

This paper focuses on the *Descriptive Ontology for Linguistic and Cognitive Engineering* (DOLCE) MasoloBorgo2003,borgoDOLCE2022, an ISO standard top-level ontology (ISO/IEC 21838). DOLCE originated as a formal theory in first-order modal logic, specifically, in the logic QS5,fitting2012first. This version is commonly referred to as the 'D18' version, after the technical report (deliverable) where it was first presented MasoloBorgo2003. 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 by [Kutz and Mossakowski2011].[2] 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.[3] 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.[4] This novel version, termed "DOLCE Simple"

[1] https://www.w3.org/TR/owl2-overview/

[2] For an implementation of this strategy, see also https://github.com/spechub/Hets-lib/tree/master/Ontology/Dolce

[3] 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

[4] https://www.iso.org/standard/78927.html

is available online.[5]

The format of this version includes the rendering of DOLCE Simple in Common Logic (CLIF[6]), as well as in the usual input formats for theorem prover (tptp, Prover9). The theory does not use the additional features of Common Logic that extend first-order logic, it is indeed a first-order theory. Therefore, throughout this paper we denote this theory by $\text{DOLCEsimple}_{\text{FOL}}$.

The objective of $\text{DOLCEsimple}_{\text{FOL}}$ is to enable automated theorem proving and standard model finders (such as Mace4) to be capable of returning at least a "small model" of DOLCE.[7] Other foundational ontologies follow suit to facilitate the application of provers and model finders, such as BFO otte2022bfo, TUPPER gruninger2022tupper, and UFO guizzardi2022ufo. For an overview and test of these implementations, see the work of [Garbacz2022].

A primary requirement of our novel OWL 2 version of DOLCE is the 'compatibility' with the (first-order logic) version $\text{DOLCEsimple}_{\text{FOL}}$. Compatibility here means, in practice, that the translation into first-order logic of the axioms of the OWL 2 version of DOLCE can be proved from the (first-order) version $\text{DOLCEsimple}_{\text{FOL}}$. That is, every model of $\text{DOLCEsimple}_{\text{FOL}}$ is also a model of the OWL 2 version. Although the OWL 2 version is less specific, as expected, and allows for many more, possibly unintended, models, the intended models of the original ontology are preserved. This result was achieved for $\text{DOLCEsimple}_{\text{FOL}}$ and a preliminary OWL 2 version of DOLCE: the proof that $\text{DOLCEsimple}_{\text{FOL}}$ entails the OWL 2 version was done by means of Prover9[8].

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 presutti2016dolce; see [Borgo et al.2022] for a chronology of previous OWL versions of DOLCE. None of these proposals adopted the same objective as ours, sticking as close as possible to the original DOLCE in FOL, nor did they prove that their OWL versions are entailed by the FOL version of DOLCE (which in some cases, it is clearly not the case). A preliminary OWL 2 version of DOLCE Simple was also developed for the

[5] `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`.

[6] `https://www.iso.org/standard/66249.html`

[7] For the full list of simplifications of $\text{DOLCEsimple}_{\text{FOL}}$ w.r.t D18 see the documentation at `http://www.loa.istc.cnr.it/index.php/dolce`.

[8] See footnote 4.

inclusion in ISO 21838 standard[9].

In this paper, starting from the OWL 2 version developed for the ISO 21838, we document our advancement in the project of approaching DOLCE with OWL 2[10]. The main motivation for the present novel proposal is driven by the intention to present DOLCE in OWL 2 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 extensions, also in OWL 2[11]. For this reason, we will explore the conceptual expressivity of OWL 2 to capture, as far as possible, the original spirit of the ontological analysis presented in the D18.

In this paper, we introduce and discuss a core theory, here termed "DOLCEbasic$_{\mathrm{OWL}}$", 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. We also discuss the modular approach and the strategy adopted to extend this core module. We then present the additional module extending this core ontology with the n-ary relations ($n > 2$) of DOLCE —i.e. ternary and quaternary relations— here termed "DOLCEnaryRel$_{\mathrm{OWL}}$". This extension includes the temporalised relations of DOLCE (e.g. temporary parthood, constitution, participation, etc.) as well as a number of mereological relations (e.g. binary sum). The use of the resulting theories, DOLCEbasic$_{\mathrm{OWL}}$ and DOLCEnaryRel$_{\mathrm{OWL}}$, is illustrated with examples taken from the literature. All the theories discussed in this paper are available online.[12]

The remainder of this paper is organised as follows. Section 2 presents the overall approach to develop the OWL 2 version of DOLCE. Section 3 illustrates the formalisation in OWL 2. The proposed ontology is tested by automatically proving its axioms from the axioms of DOLCEsimple$_{\mathrm{FOL}}$ in Section 4 and is exemplified via a use case in Section 5. Finally, Section 6 draws the conclusions. The appendix lists the OWL 2 entities of our ontologies, summarises their intended meaning, and indicates the correspondence with the original D18 entities.

[9] See http://www.loa.istc.cnr.it/index.php/dolce.

[10] This paper is a substantial extension of what presented by [Porello et al.2024].

[11] Note that we do not use the term 'module' to indicate logically independent subtheories.

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

2 Approach

2.1 Goal

Our goal is to develop an OWL 2 version of DOLCE that is close to and justified by the original theory presented in D18 MasoloBorgo2003, and simplified in DOLCEsimple$_{\mathrm{FOL}}$. Specifically, we are working with a fragment of the Description Logic $\mathcal{SROIQ}$, a sub-logic of OWL 2 HorrocksKS06

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 $n > 2$ and a structural restriction on theories, specifically on the *role box*, which demands *regularity* and *simplicity* HorrocksKS06. Regularity demands only simple dependencies between roles in the role hierarchy and is a structural property of the role box.

Methodologies for approximating first-order theories with $\mathcal{SROIQ}$ have been developed for instance by [Benevides et al.2019] and [Hahmann and Powell II2021]. The strategy by [Benevides et al.2019] involves checking a large number of candidate $\mathcal{SROIQ}$ 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 formally assess the proximity of our OWL 2 version to the original FOL-DOLCE. In fact, there is no unique closest approximation to DOLCE and, as [Benevides et al.2019] experiment effectively show, choosing among the many close approximations is often impractical. Instead, we will select the axioms or theorems from the D18 version that we wish to keep in OWL 2, focusing on the practical use and general understandability of such constraints and axioms. The intended application, besides providing a consistent, decidable, and well-motivated version of DOLCE, is to facilitate its use as an upper-level ontology for domain ontologies. These domain ontologies can be linked to our OWL version of DOLCE to verify their consistency with its core assumptions.

In this article, we will systematically use a first-order syntax for all formulas, be they intended to be part of a first-order theory such as the original DOLCE in D18, or a part of an OWL theory. The vocabulary is mostly that of the OWL theories proposed, but we sometimes refer to the original DOLCE vocabulary. See the Appendix for a table aligning the two vocabularies. We will use a `typewriter font` for all logical formulas and fragments thereof.

2.2 A modular approach

An obvious constraint when approximating a first-order theory in OWL 2 is that all standard 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 ($\text{DOLCEbasic}_{\text{OWL}}$) which is limited 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 $\text{DOLCEbasic}_{\text{OWL}}$ as the object property (i.e. binary) `constantParticipantOf`. Another module, termed $\text{DOLCEnaryRel}_{\text{OWL}}$, which extends (imports) $\text{DOLCEbasic}_{\text{OWL}}$, is dedicated to properly introduce and treat the n-ary relations (for $n > 2$) in OWL 2. The adopted modular approach allows for adding still other modules to extend $\text{DOLCEbasic}_{\text{OWL}}$ with core concepts or domain-level concepts, possibly independent or dependent on each other.

In designing the module $\text{DOLCEnaryRel}_{\text{OWL}}$, we follow the "reification" guidelines advocated by the W3C[13], inspired by the neo-Davidsonian approach to handling events and their arguments in natural language semantics Parsons1990. Several other approaches, with and without reification, to deal with temporalised relations in OWL have been discussed garbacz2017representation. 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", a sort of 4D approach to endurants. This approach had been proposed earlier for an ontology of "fluents" in OWL weltyfikes2006fluents. Handling ternary temporalised relations directly as binary ones in OWL has the considerable advantage of enabling the expression of transitivity and other properties supporting reasoning, which is extremely limited with the standard W3C approach, as we will also see below. Unfortunately, the Temporally Qualified Continuants Pattern approach cannot be generalised in OWL to non-temporalised ternary relations, such as the mereological sum

[13] `https://www.w3.org/TR/swbp-n-aryRelations`

between abstracts, nor to temporalised relations of arity above 3, such as the temporalised mereological sum between endurants.

The approach to n-ary relations ($n > 2$) followed here implies the introduction of a new category of entities called "reified relationships" (not included in DOLCEbasic$_{\text{OWL}}$), to which each argument of the original relation will be related, by a distinct new binary relation, namely, an OWL object property. For example, the mereological sum[14] between perdurants or abstracts is defined by a ternary relation `sum(x,y,z)` indicating that the sum of `x` and `y` results in `z`. Such mereological sums will be represented in DOLCEnaryRel$_{\text{OWL}}$ by the following assertions, where `RRelSum` stands for "reified relationship of sum":

```
RRelSum(r)

sumAddend1(r,x)

sumAddend2(r,y)

sumResult(r,z)
```

The class `RRelSum` will be then a subclass of the class `ReifiedRelationship`.

2.3 Ontology of reified relationships

An important ontological question regards the nature of these reified relationships: whether this new category fits within the original DOLCE taxonomy, and if so, under which main category, and if not, how to handle it. We now discuss the two most apparently reasonable options for categorising the reified relationships as particulars in DOLCE. As mentioned above, our goal is to find a solution that does not impose to revise significantly the original DOLCE (D18).

These two options have in common to view a reified relationship as some sort of "state of the relation holding" aka "situation"[15] 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; perdurants are disjoint with facts, which are outside of time and thus abstracts in DOLCE).

In the first option, reified relationships as perdurants, the reification of a ternary relation `r(a,b,c)` would mean "the (event or state of) holding of

[14]The binary operator + introduced in the D18 report can be equivalently encoded as a ternary relation, as usual.

[15]See, e.g., `https://nemo-ufes.github.io/gufo/#situations`

relation r among a, b, and c". This would be in line with the neo-Davidsonian approach to events, and seems quite natural for temporalised relations and perhaps for ternary relations between perdurants. However, the approach 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 outside of time in DOLCE. What would we do with these non temporalised reified relationships, and why not handling all of the reified relationships in a unified way since the motivation to introduce them stems from a same issue, namely, the limited expressivity of OWL?

Furthermore, a delicate aspect arises when considering the axiom of DOLCE which states that all perdurants have endurants that participate to them at some time. If we consider relationships as perdurants, they must satisfy to this axiom too, potentially leading to a regress. As our goal is to adhere as closely as possible to the original DOLCE, modifying this axiom to accommodate a new subclass of perdurants seems undesirable. On the basis of these considerations, and disregarding for the moment the non-homogeneity in handling reified relationships just pointed at, we shall now examine more closely if there is an alternative manner to keep this axiom, and yet include some reified relationships as perdurants while avoiding any regress.

Suppose we intend to reify the participation of an endurant a to an event e at time t, i.e. pc(a,e,t). The relational statement is then reified by means of a perdurant e_1, with projections $p_1(e_1,a)$, $p_2(e_1,e)$, $p_3(e_1,t)$ that relate e_1 to the arguments of the statement. Since e_1 is also a perdurant, according to DOLCE, it must be present at least at one time interval t_1 (cf. Td15[16]) and it must have at least a participant (cf. Ad34), say a_1. Therefore, our theory has $pc(a_1,e_1,t_1)$ as consequence. Again, we reify the relationship by means of a perdurant, say e_2, which requires at least one participant, say a_2, so that $pc(a_2,e_2,t_2)$ holds in our theory. Following this option, we are led to populate the ontology with an unbounded number of perdurants.

One way to stop the regress is to force that the perdurant e_1 (the reification of the relationship pc(a,e,t)) is the same as the perdurant occurring in the participation relation, i.e. $e = e_1$. However, consider the example: "Mary participates in the football match f at t", i.e. pc(m,f,t). According to this requirement, the football match f has to be the same event as "the participation of Mary to the football match f at t", which is not only odd, but would be indistinguishable from the participation of Mary to the football

[16]We here refer to theorems (Tdx), axioms (Adx) and definitions (Ddx) of DOLCE as in the D18 technical report.

match at some other time $\mathtt{t}_1$ and from the participation of Lea to the same football match at $\mathtt{t}$ or $\mathtt{t}_1$. To try to avoid this, one could force that the reification event is only a part of the event $\mathtt{e}$ occurring in $\mathtt{pc(a,e,t)}$. In our example, "the participation of Mary to the football match $\mathtt{f}$ at $\mathtt{t}$" should be part of $\mathtt{f}$, which is more plausible. So, given $\mathtt{pc(a,e,t)}$, we would have to impose that its reification is a perdurant $\mathtt{e}_1$ which is a part of $\mathtt{e}$. Then, we have to impose that the participants of $\mathtt{e}_1$ at $\mathtt{t}_1$ are parts of some participants of $\mathtt{e}$, say $\mathtt{p(a_1,a,t_1)}$, and that the reification of $\mathtt{pc(a_1,e_1,t_1)}$ is in turn part of $\mathtt{e}_1$. Still, this option does not prevent an unbounded addition of perdurants, unless we assume that there is a minimal (wrt. parthood) event with some participant $\mathtt{a}_i$. The existence of minimal events (atomicity of perdurants) is not assumed in DOLCE, so this move would entail a significant revision of the original views of DOLCE.

The second option supposes that the reified relationships are abstracts of DOLCE. For instance, the participation $\mathtt{pc(a,e,t)}$ is reified as the "fact" of $\mathtt{a}$ being a participant of $\mathtt{e}$ at $\mathtt{t}$. In this case, we view facts as abstract entities (as in D18), outside of time, so the previous perdurant regress is blocked.

However, consider the ternary relational statement $\mathtt{sum(a,b,c)}$, i.e. "$\mathtt{a}$ is the sum of $\mathtt{b}$ and $\mathtt{c}$". We reify the relationship as an abstract $\mathtt{d}$. We have now two options. The first one (*i*), $\mathtt{d}$ is the same entity as $\mathtt{a}$, that is "the fact that $\mathtt{a}$ is the sum of $\mathtt{b}$ and $\mathtt{c}$" is identified with the the sum $\mathtt{a}$ of $\mathtt{b}$ and $\mathtt{c}$. The second one (*ii*), $\mathtt{d}$ is not the same entity as $\mathtt{a}$. Case (*i*) is counterintuitive for general abstract entities. Given two regions (which are abstracts in DOLCE), their sum is a region and not the "fact that a certain region is the sum of two other regions". Thus, we have to exclude (*i*), or to properly constrain the reified relationships to specific subclasses of abstracts (e.g. the reification of the sum of facts is a fact, the reification of the sum of regions is a region, etc.). But no similar move could apply to reified relationships not involving abstracts, like, e.g., the ternary sum between perdurants, and we would in this case too have a non-homogeneous handling of n-ary relations.

In case (*ii*), a regress is triggered again, given that in DOLCE the sum of abstracts always exists. If $\mathtt{d}$ is the reification of $\mathtt{sum(a,b,c)}$, and $\mathtt{d} \neq \mathtt{a}$, DOLCE infers that $\mathtt{a}_1$ exists, where $\mathtt{sum(a_1,d,a)}$, so we reify it by $\mathtt{d}_1$. Then $\mathtt{a}_2$ exists, such that $\mathtt{sum(a_2,d_1,a)}$, which is reified by $\mathtt{d}_2$, and so on. One could object that mereology among abstracts should not be uniform but rather be segregated in different subclasses so the sum of a region and a fact doesn't exist. However, a regress is still obtained by considering sums of reified relationships, that is, sums among facts. If $\mathtt{d}_1$ and $\mathtt{d}_2$ are two distinct reified relationships then their sum $\mathtt{d}_3$ exists such that $\mathtt{sum(d_3,d_1,d_2)}$ which is reified into $\mathtt{d}_4$, and so on. Just as before, there appears to be no way to ensure that $\mathtt{d}_4$ is identical to $\mathtt{d}_3$ and block the regress.

But if there was a way without a deep modification of DOLCE axioms, there is yet another, final, consideration to make. Adopting the option to categorise reified relationships as abstracts is a strategy that gives an ontological importance to some reified relationships which had no place in the original D18 DOLCE. This strategy implies a change of perspective on DOLCE itself, while here we try to stick as closely as possible to the original DOLCE. Indeed, with such a change of perspective, why reifying only ternary and quaternary relations but not binary relations? For instance, if there are ontological grounds to sustain that the "fact" of a ternary `partOf` (called `P` in D18) relation holding between two endurants at a time should exist and be introduced as a reified relationship, then one should find an explanation why this doesn't apply to the "fact" of a binary `partOf` relation holding between two perdurants, and for all other binary relations, even those involving time such as `presentAt` and `temporallyLocatedAt`. As far as we know, no-one trying to deal with n-ary relations ($n > 2$) in an OWL ontology is promoting the solution of reifying all relations, including binary ones. This is for a good reason, as it would lead to renouncing most of the inferential power of description logics over binary relations. But again, from an ontological point of view, having a non-homogeneous handling of relations cannot be justified.

For the sake of completeness with respect to the main categories of DOLCE, we conclude our analysis by discussing the possibility of viewing reified relationships as either endurants or qualities of DOLCE.

Let us first consider the option of viewing reified relationships as endurants. For example, `pc(a,e,t)` is represented by an endurant $\mathtt{a}_1$. Notice that $\mathtt{a} \neq \mathtt{a}_1$, otherwise the same entity would be the reification of all the participations that `a` entertains with any perdurant. By axiom (Ad34), $\mathtt{a}_1$ must participate to at least one perdurant, say $\mathtt{e}_1$. Thus we have $\mathtt{pc(a}_1\mathtt{,e}_1\mathtt{,t)}$, which requires a new endurant $\mathtt{a}_2$ as reification. Notice that it is implausible, again, to state that $\mathtt{e} = \mathtt{e}_1$. For example, "Mary participates in the football match `f` at `t`" would be reified as an endurant $\mathtt{a}_1$, which participates to at least one perdurant $\mathtt{e}_1$. If $\mathtt{f} = \mathtt{e}_1$, the football match `f` must have, among its participants, the 'participation of Mary to `f`', which is conceptually disturbing. Thus, categorising reified relationships as endurants leads again to admitting an unbounded number of endurants and perdurants.

Finally, reified relationships are not qualities of DOLCE. Among many reasons for not pursuing this strategy, each quality inheres in a single entity, see (Ad46)–(Ad48), thus in DOLCE, we cannot associate a single quality, e.g., to a triple of entities with the aim of reifying a ternary relationship.

We therefore conclude that integrating reified relationships as a subcategory

in the original DOLCE taxonomy is not possible—neither under perdurants, nor under abstracts, nor under endurants or qualities. However, adding a new, ontologically meaningful, category to the top-level of DOLCE would be significantly departing from the original DOLCE vision on how reality is carved. In addition, such a new category would require, to follow the methodology used to build DOLCE, a strong philosophical motivation and explanation of what exactly these entities are, which we believe is a non-trivial task.
Our conclusion is that there is no alternative to handling reified relationships as mere technical additions to the ontology, unrelated with the original rationale of the taxonomy. One is forced to introduce such technical devices in OWL versions of DOLCE for technical reasons, because of the limited expressivity of $\mathcal{SROIQ}$, and not for genuine ontological reasons.

2.4 Consequences on the taxonomy of reified relationships

Since reified relationships cannot be seen in any way as a meaningful "state or fact of the relation holding", they are to be seen as mere technical additions to circumvent expressivity limitations. They are then treated in this paper as purely syntactic devices, and thus appear in the hierarchy of classes in a completely separated branch from DOLCE's taxonomy, so the branch dominated by `ReifiedRelationship` and the branch dominated by `Particular` (the top class of DOLCE) are totally disjoint.

Another important consequence is that a given reified relationship is uniquely related to a single atomic formula, i.e., a given reified relationship instantiates a single subclass of `ReifiedRelationship` corresponding to one n-ary relation ($n > 2$) of DOLCE, and is related to a unique tuple of arguments through (more or less) specific binary relations, all subrelations of the general binary relation `hasArgument`. This means that all the reified relationship classes corresponding to an n-ary relation of DOLCE are disjoint. Unfortunately, this implies that the taxonomy cannot be used for reasoning over n-ary relations. For instance, we do not have `RRelTempPart(x)` $\rightarrow$ `RRelTempOverlap(x)` to relate temporary part to temporary overlap. Moreover, as we will see below, there is no alternative way to recover such an inference given the expressivity of OWL.

3 OWL 2 Formalization

This section illustrates and motivates the main features of the core modules DOLCEbasic$_{\text{OWL}}$ and DOLCEnaryRel$_{\text{OWL}}$ formalised in OWL 2.[17] OWL entities (e.g., classes and object properties) are represented using their local IRI in typewriter style. To improve readability, long local IRIs may be displayed broken into separate lines in the figures. Prefixes are generally omitted unless required to avoid ambiguity. Table 1 lists the relevant prefixes along with their corresponding namespaces. Appendix A provides the complete lists of OWL entities with their prefix, label (`rdfs:label`), comment (`rdfs:comment`), and acronym used in D18.

Figure 1 shows the two top classes of the DOLCEbasic$_{\text{OWL}}$ and DOLCEnaryRel$_{\text{OWL}}$ modules considered as a whole, i.e. `Particular` and `ReifiedRelationship`, respectively. DOLCEbasic$_{\text{OWL}}$ exploits only the `Particular` branch while DOLCEnaryRel$_{\text{OWL}}$ exploits both branches.

The following subsections delve into the class and property hierarchies in DOLCEbasic$_{\text{OWL}}$ (Sect.3.1), the modeling of endurants, perdurants, and qualities (Sect.3.2), the modeling of time and mereology (Sect.3.3), additional classes and properties in the hierarchies for DOLCEnaryRel$_{\text{OWL}}$ (Sect.3.4), the argument structure of reified relationships (Sect. 3.5), and some axioms that can and cannot be expressed in DOLCEnaryRel$_{\text{OWL}}$ (Sect. 3.6).

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#`
dn:	`https://w3id.org/DOLCE/OWL/DOLCEnaryRel#`

[17]More precisely, our OWL formalisation requires only a sublogic of $\mathcal{SROIQ}$, namely the logic $\mathcal{SRIF}$. The logic used by the ontology can also be verified using an ontology metrics generator, such as the one available at `https://ontometrics.informatik.uni-rostock.de/ontologymetrics/`. The logic $\mathcal{SRIF}$, although decidable, is computationally very demanding, as usual for expressive fragments of Description Logics. For an analysis of computational complexity of $\mathcal{SRIF}$, see [Horrocks and Sattler2004]. We leave a comprehensive analysis of the complexity and of the efficiency of our rendering to future work, but see our illustrative cases in Section 5.

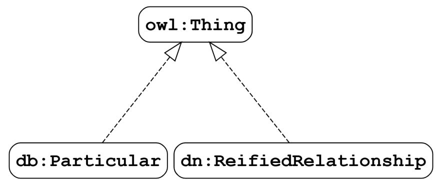

Figure 1: The two most general classes of DOLCEbasic$_{\text{OWL}}$ and DOLCEnaryRel$_{\text{OWL}}$. Rounded boxes represent classes identified by their local IRIs with prefixes. Dashed arrows represent `rdfs:subClassOf` relations.

3.1 Class and Property Hierarchy in DOLCEbasic$_{\text{OWL}}$

The taxonomy of DOLCEbasic$_{\text{OWL}}$ is presented in Figure 2. There are only few differences with respect to the taxonomy of DOLCE as depicted in D18: `Atom` and `ConstantAtom`, hereby included, are in fact not new classes, since they are defined in DOLCE; they are not displayed in the standard DOLCE taxonomy just because they are not disjoint with their siblings.[18]

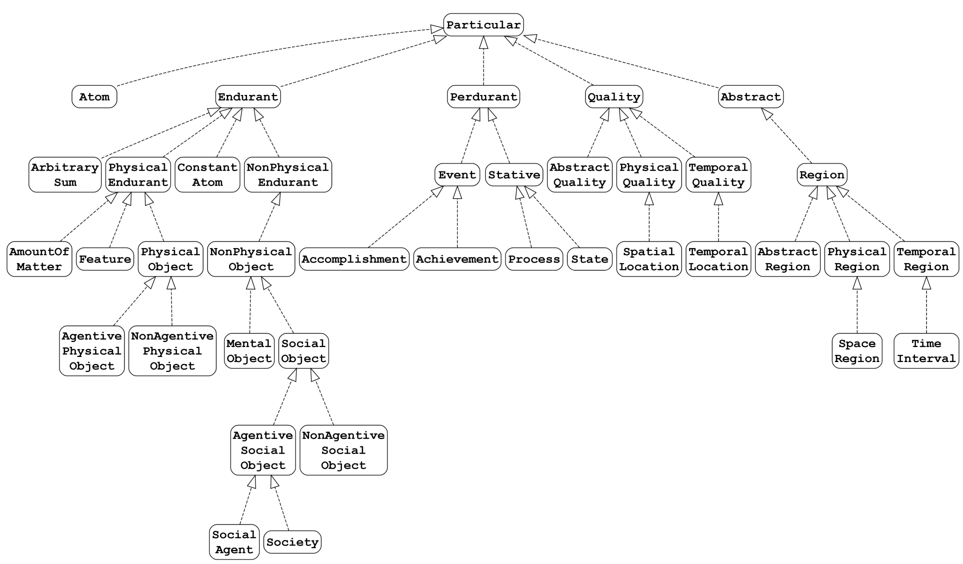

Figure 2: DOLCEbasic$_{\text{OWL}}$ class hierarchy. Rounded boxes represent classes identified by their local IRI. Prefix `db:` is omitted for all classes. Dashed arrows represent `rdfs:subClassOf` relations.

[18] A definition of these classes is included in DOLCEsimple$_{\text{FOL}}$, see `https://github.com/appliedontolab/DOLCE/tree/main/OWL/Proof`.

The hierarchy of the object properties in DOLCEbasic$_{\mathrm{OWL}}$ is presented in Figure 3. Notice the properties corresponding to 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. As explained above, our strategy is to include in the object properties of DOLCEbasic$_{\mathrm{OWL}}$ the constant versions of all the temporalised ternary relations of DOLCE, which are then binary only. For instance, the ternary parthood relation between endurants in DOLCE is accounted for in DOLCEbasic$_{\mathrm{OWL}}$ only as its constant version `constantPartOf`, with the following "definition" (definitional axiom) matching the one in DOLCE (D18):

$$\texttt{constantPartOf(x,y)} \leftrightarrow (\exists t \texttt{presentAt(y,t)} \wedge \forall t\ (\texttt{presentAt(y,t)} \rightarrow (\texttt{presentAt(x,t)} \wedge \texttt{partOf(x,y,t)})))$$

This definition implies that when `x` is a constant part of `y`, the time interval in which the whole `y` is temporally located (its temporal extension, that is, the time of its whole "life") is included in that of the part `x`. Here and for other properties that are constant versions of temporalised relations, the temporal extensions of the arguments are not necessarily identical (for more details, see Sections 3.3 and 4.1).

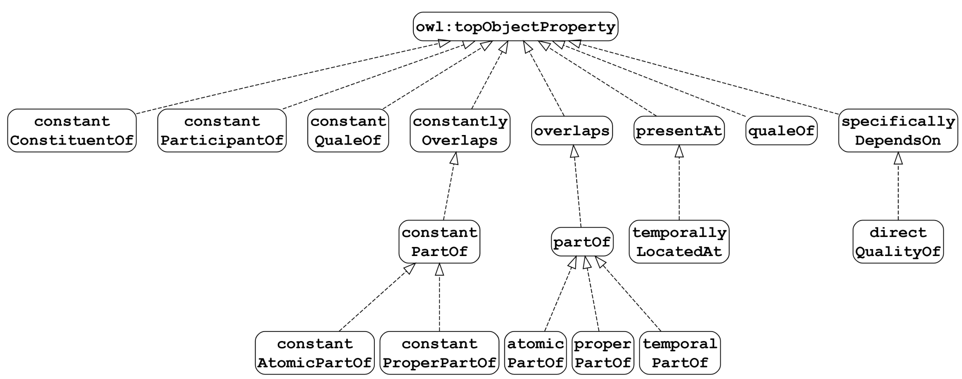

Figure 3: DOLCEbasic$_{\mathrm{OWL}}$ property hierarchy. Rounded boxes represent object properties identified by their local IRI, while prefix `db:` is omitted. Dashed arrows represent `rdfs:subPropertyOf` relations.

3.2 Focus on Endurants and Perdurants

Figure 4 shows some of the object properties having classes `Endurant` and `Physical Endurant` as their domain or range in

DOLCEbasic$_{\text{OWL}}$. These properties correspond to *parthood* (constantPartOf, constantProperPartOf), *constitution* (constantConstituentOf), *presence* (presentAt, temporallyLocatedAt), *participation* (constantParticipationOf), and *quality* characterization (directQualityOf).

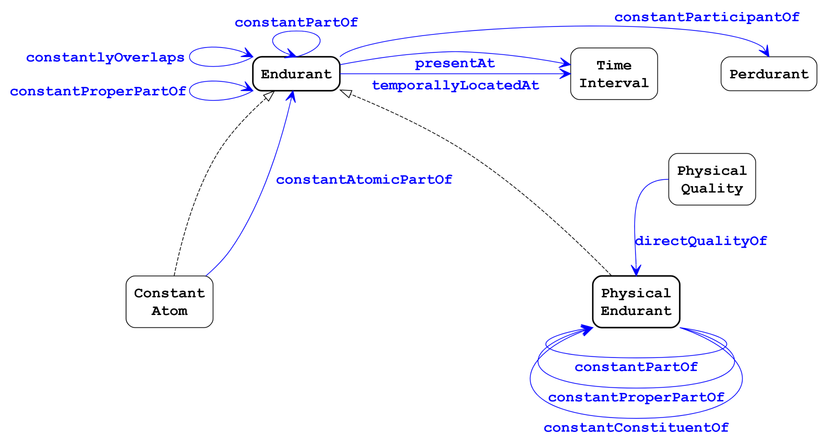

Figure 4: Object properties for Endurant and PhysicalEndurant in DOLCEbasic$_{\text{OWL}}$. Rounded boxes represent classes identified by their local IRIs. Blue solid arrows denote object properties identified by their local IRIs, linking classes in their rdfs:domain to classes in their rdfs:range. Dashed arrows represent rdfs:subClassOf relations. The prefix db: is omitted for all classes and object properties.

Figure 5 primarily illustrates the object properties having class Perdurant as their domain or range, including those corresponding to *parthood* (partOf, properPartOf, temporalPartOf), *presence* (presentAt, temporallyLocatedAt), *participation* (constantParticipationOf), and *quality* characterization (directQualityOf). The property temporalPartOf is the sub-property of partOf among perdurants in which the part is a "temporal slice" of the whole, that is, a maximal part of the whole during some time interval, as e.g., the first set of a tennis match. One can note that to facilitate the direct reference to the time intervals that are qualia of the TemporalLocation qualities of perdurants, that is, the maximal time intervals at which they are present, called temporal extensions, the property temporallyLocatedAt can be used over perdurants as well as over endurants and qualities.

Figure 6 illustrates the important role individual qualities play in DOLCE. The class AbstractQuality is dedicated to qualities that are linked through `directQualityOf` to their bearers that are `NonPhysicalEndurant` entities (and not to `Abstract` entities), `PhysicalQuality`, among which the subclass of `SpatialLocation`, to `NonPhysicalEndurant` entities, and `TemporalQuality`, among which the subclass of `TemporalLocation`, to `Perdurant` entities. In DOLCEbasic$_{\mathrm{OWL}}$, individual qualities are linked to their constant qualia belonging to corresponding subclasses of `Region` through the object property `constantQualeOf`. Qualities whose qualia vary in time can be described only in DOLCEnaryRel$_{\mathrm{OWL}}$. As described in the next section, temporal qualities of perdurants such as those of the class `TemporalLocation` do not have qualia that vary in time, so the property `qualeOf` is directly corresponding to the DOLCE relation.

3.3 Focus on Time and Mereology in DOLCEbasic$_{\mathrm{OWL}}$

DOLCE provides a rich representation of how entities behave through time, and this attention to time is present in this OWL 2 rendering although in DOLCEbasic$_{\mathrm{OWL}}$ we only have binary constant versions of DOLCE relations. Here we intend to highlight how some of those temporal aspects are encoded on DOLCEbasic$_{\mathrm{OWL}}$, what could be implemented, and what could not.

As sketched in Figures 5-6, a perdurant has temporal qualities (e.g. the duration of an event), among which a unique quality in the class `TemporalLocation`. The quality `TemporalLocation` 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 binary relations, among which the `partOf` object property in DOLCEbasic$_{\mathrm{OWL}}$. They can be atomic and non-convex, i.e., scattered through time (as generalised intervals of [Ligozat1991]). As noted above, the perdurant is also directly related to the same time interval of its whole temporal extension through the property `temporallyLocatedAt`. Also endurants and qualities are temporally located at a time interval, their temporal extension.[19] The property `presentAt` relates perdurants, endurants and qualities to any time interval that is part of their whole temporal extension.

Although the expressivity of OWL 2 doesn't enable a proper characterization of mereological relations[20] and despite the simplicity and the regularity constraints on properties in OWL 2, we were able to make some choices and

[19] 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".

[20] This fact is well-known KeetKutz17.

guarantee a few facts.

We opted to keep the transitivity of partOf and properPartOf and to drop the irreflexivity of properPartOf. We had also to drop the reflexivity of partOf because 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, we would need to use of the $\exists$R.Self $\mathcal{SROIQ}$ constructs, which requires a simple property R and is thus unfortunately incompatible with transitivity. Antisymmetry is impossible to express. partOf effectively 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 could 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. Here and elsewhere, users of DOLCEbasic$_{\mathrm{OWL}}$ should be aware of the intended semantics of all the vocabulary to make sure the Aboxes of their knowledge bases effectively respect the vision of DOLCE.

As explained in Section 2.2, 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, as mentioned above, and the substrate is present when the entity it constitutes is present (see proof in Section 4). When completing with object properties that are constant versions of all temporalised relations, a decision had 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 will be seen in Section 4.1, an axiom enforcing that the temporal extensions of both arguments overlap is expressible in OWL 2, 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. Again, this means the user should be aware of how to use in a coherent way such object properties of $\text{DOLCEbasic}_{\text{OWL}}$, as the expressivity of the language is too weak to exclude unintended models.

3.4 Class and Property Hierarchy in $\textbf{DOLCEnaryRel}_{\textbf{OWL}}$

The taxonomy of the proposed $\text{DOLCEnaryRel}_{\text{OWL}}$ ontology module is represented in Figure 7. Subclass relations are used to group reified relationships that have common behaviours, not to grasp inferences among DOLCE n-ary relations (see Section 2.3 for motivation). For instance, grouping all temporalised relationships under `ReifiedTemporalisedRelationship` allows to use a common object property `rRelTime` to link them to their `TimeInterval` "argument". It was also necessary to distinguish several reified relationships according to the class of their arguments where DOLCE had only one relation, in order to be able to enforce some axioms on them. For instance the sum reified relationship for abstracts had to be distinguished from the sum reified relationship for perdurants, while there was a single sum relation for both categories in DOLCE. The taxonomy allows for grouping such cases according to the original DOLCE relation.

The set of object properties in $\text{DOLCEnaryRel}_{\text{OWL}}$ is represented in Figure 8. All properties, except for `atomAt`, are sub-properties of the general `hasArgument` that links a reified relationship to one of its "arguments", that is, an argument of the corresponding n-ary relation in DOLCE. The property `rRelTime` is used to link all reified temporalised relationships to their temporal argument. All other properties used to describe the argument structure of reified temporalised relationships have their names prefixed with `temp`. When a semantic distinction could be made among arguments, the properties were named accordingly, as with `tempPart` and `tempWhole`, otherwise those properties included a reference to the name of the original DOLCE relation and a number to reflect the order of its arguments as in D18, as with `tempOverlapArg1` and `tempOverlapArg2`.

The property `atomAt` directly corresponds to a binary relation of DOLCE. It is included in $\text{DOLCEnaryRel}_{\text{OWL}}$ and not in $\text{DOLCEbasic}_{\text{OWL}}$ because it belongs to the subtheory for the temporalised mereology among endurants, which is covered in $\text{DOLCEnaryRel}_{\text{OWL}}$.

3.5 Focus on the Argument Structure of Reified Relationships in DOLCEnaryRel$_{\mathrm{OWL}}$

The following figures further illustrate the argument structure of the reified relationships used to represent temporalised mereology among endurants—except for sum (Figure 9), as well as constitution (Figure 10), participation (Figure 11), and quale-of (Figure 12) relations of DOLCE. As explained above, in order to enforce the homogeneity of arguments for reified relationships such as `RRelTempConstitution` in Figure 10, we distinguished the subclasses `RRelTempConstitutionNPED`, `RRelTempConstitutionPED` and `RRelTempConstitutionPD` dedicated to constitution among non physical endurants, physical endurants and perdurants, respectively. A similar method was used for `RRelTempConstitutionQuale` in Figure 12 and for `RRelSum` in Figure 13.

Figure 13 illustrates the different reified relationships for sum operators. Some of those correspond to binary operators, i.e. ternary relations, in DOLCE: sums among abstracts (`RRelSumAB`) or perdurants (`RRelSumPD`) and constant sums among endurants (`RRelConstantSum`). There are also sum reified relationships corresponding to ternary operators, i.e. quaternary relations, in DOLCE: temporalised sums among endurants (`RRelTempSum`).

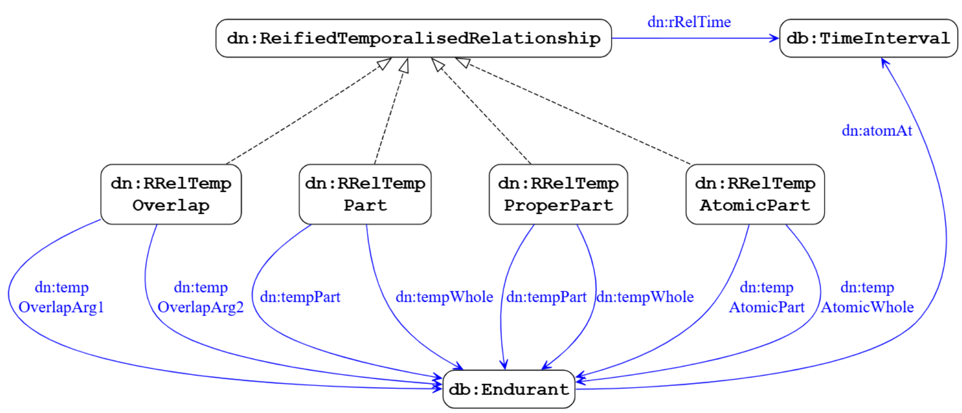

Figure 9: Reified relationships of DOLCEnaryRel$_{\mathrm{OWL}}$ for temporalised mereology among endurants. The property `atomAt` is included as it belongs to this mereology. Rounded boxes represent classes identified by their local IRI with prefix. Blue solid arrows denote object properties identified by their local IRIs with prefix, linking classes in their `rdfs:domain` to classes in their `rdfs:range`. Dashed arrows represent `rdfs:subClassOf` relations.

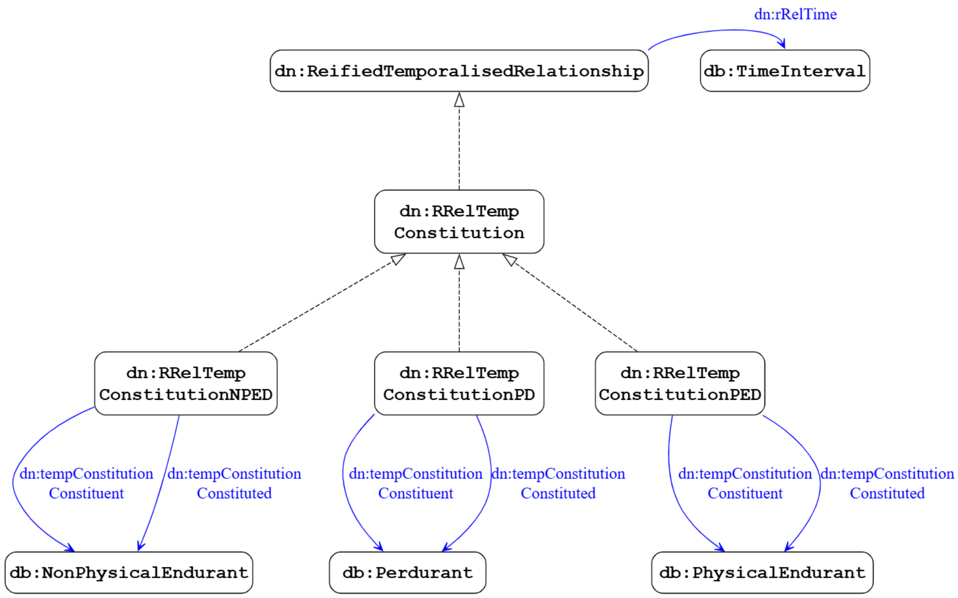

Figure 10: Reified relationtionships of $\text{DOLCEnaryRel}_{\text{OWL}}$ for constitution. Rounded boxes represent classes identified by their local IRI with prefix. Blue solid arrows denote object properties identified by their local IRIs with prefix, linking classes in their `rdfs:domain` to classes in their `rdfs:range`. Dashed arrows represent `rdfs:subClassOf` relations.

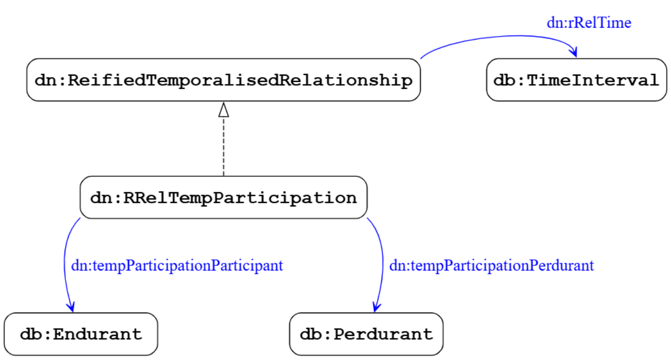

Figure 11: Reified relationships of $\text{DOLCEnaryRel}_{\text{OWL}}$ for participation. Rounded boxes represent classes identified by their local IRI with prefix. Blue solid arrows denote object properties identified by their local IRIs with prefix, linking classes in their `rdfs:domain` to classes in their `rdfs:range`. Dashed arrows represent `rdfs:subClassOf` relations.

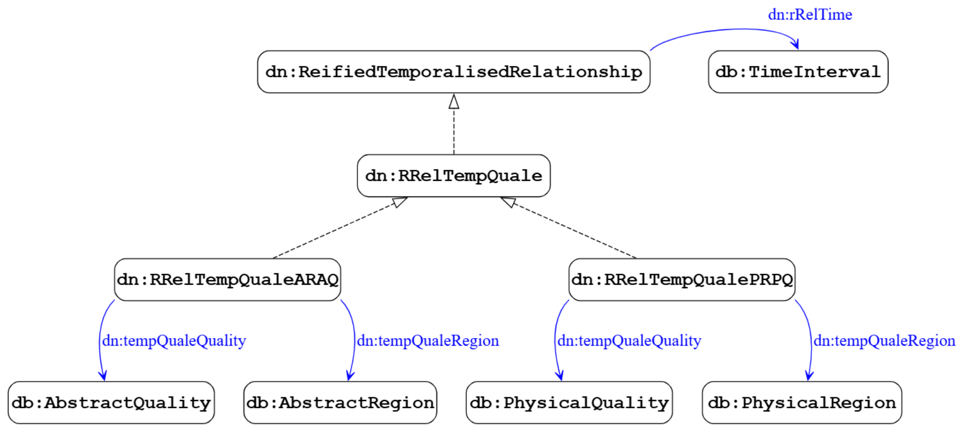

Figure 12: Reified relationships of DOLCEnaryRel$_{\mathrm{OWL}}$ for quale-of. Rounded boxes represent classes identified by their local IRI with prefix. Blue solid arrows denote object properties identified by their local IRIs with prefix, linking classes in their `rdfs:domain` to classes in their `rdfs:range`. Dashed arrows represent `rdfs:subClassOf` relations.

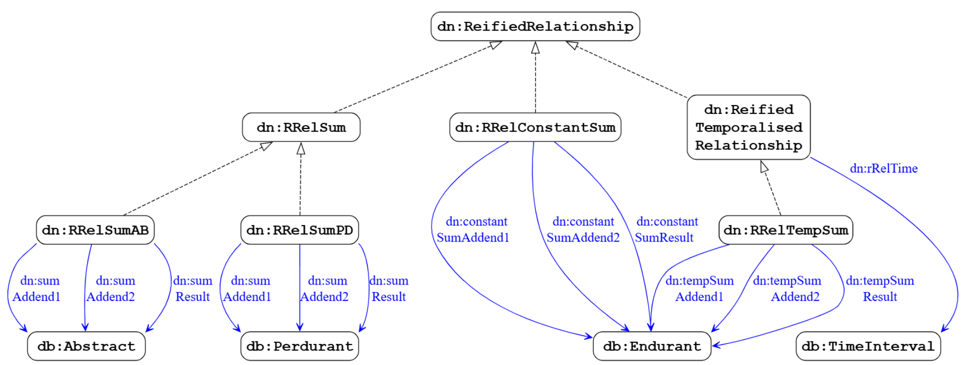

Figure 13: Reified relationships of DOLCEnaryRel$_{\mathrm{OWL}}$ for sum. Rounded boxes represent classes identified by their local IRI with prefix. Blue solid arrows denote object properties identified by their local IRIs with prefix, linking classes in their `rdfs:domain` to classes in their `rdfs:range`. Dashed arrows represent `rdfs:subClassOf` relations.

3.6 Focus on some Axioms in DOLCEnaryRel$_{\mathrm{OWL}}$

As anticipated above, the reified relationship approach doesn't allow to characterise much of their inferential behaviour, as the expressivity of OWL

2 is limited. It is not possible, for instance, to ensure, from the existence of a reified relationship r in RRelTempProperPart which is related to its "arguments" x, y and t, that two other reified relationships exist in the classes RRelTempPart and RRelTempOverlap, that are related to the same arguments x, y and t. This means that simple axioms or theorems of DOLCE that guarantee that proper part entails part, which in turn entails overlap, are not expressible. Similarly, parthood transitivity and overlap symmetry are lost, because such axioms would imply asserting the existence of reified relationships related to specific arguments, something impossible in OWL 2. As a result, $\text{DOLCEnaryRel}_{\text{OWL}}$ cannot be used for reasoning as one would wish, and users must be careful to avoid populating this ontology in a way that departs from the original DOLCE vision. Whenever possible, that is, when time stamping and representing change are not crucial issues for the problem at hand, we encourage the use of the constant versions of these relations, which were introduced in $\text{DOLCEbasic}_{\text{OWL}}$ precisely to enable reasoning.

Nevertheless, domain and range restrictions and class disjointness can be expressed in $\text{DOLCEnaryRel}_{\text{OWL}}$. In addition, some property chains can be written to guarantee some interesting behaviour of the reified relationships in time. In particular, we can ensure that the endurants that are arguments of a RRelTempProperPart or RRelTempPart reified relationship are present during the time of the relationship, with the following property chains:

$$(\texttt{tempPart(y,x)} \land \texttt{rRelTime(y,t)}) \rightarrow \texttt{presentAt(x,t)}$$

$$(\texttt{tempWhole(y,x)} \land \texttt{rRelTime(y,t)}) \rightarrow \texttt{presentAt(x,t)}$$

Similar property chains also ensure the presence at the time of the relationship of the other (non-temporal) arguments of all instances of ReifiedTemporalisedRelationship. But unfortunately, we cannot enforce that the temporal quale of an endurant (its life) is the sum of the times of the perdurants to which the endurant participates.

Finally, some cardinality constraints have been imposed to enforce that reified relationships correspond to a unique tuple of arguments. For instance, constraints for RRelTempParticipation and RRelTempConstitution impose that they have exactly 3 arguments, while constraints for some mereological relations like RRelTempOverlap impose a maximum of 3 arguments, since overlap is reflexive. On the other hand, there is no way to guarantee the unicity of reified relationships instantiating a given leaf subclass and having the same tuple of arguments weltyfikes2006fluents.

4 Adequacy to dolce

In this Section, we establish the adequacy of the OWL 2 modules by proving the following two facts. Firstly, we show that the first-order logic $\text{DOLCEsimple}_{\text{FOL}}$ entails each axiom of $\text{DOLCEbasic}_{\text{OWL}}$. Secondly, we enrich $\text{DOLCEsimple}_{\text{FOL}}$ by adding the taxonomy of reified relationships and the all the axioms that establish that, whenever a certain reified relationship exists, a corresponding relational statement of $\text{DOLCEsimple}_{\text{FOL}}$ holds. We term this enriched version $\text{DOLCEsimpleNary}_{\text{FOL}}$. Then, we prove that $\text{DOLCEsimpleNary}_{\text{FOL}}$ entails each axiom of $\text{DOLCEnaryRel}_{\text{OWL}}$.

To prove such facts, we shall translate the OWL 2 axioms of $\text{DOLCEbasic}_{\text{OWL}}$ and $\text{DOLCEnaryRel}_{\text{OWL}}$ into first-order logic; then we apply automatic theorem proving (viz. Prover9[21] and Vampire [22]) to verify each entailment.

The first fact shows that each model of $\text{DOLCEsimple}_{\text{FOL}}$ is also a model of $\text{DOLCEbasic}_{\text{OWL}}$, therefore, the intended models of DOLCE are at least preserved by our OWL rendering. The second fact establishes that, firstly, by extending DOLCE with reified relationships, we obtain a theory which is still consistent with the original DOLCE theory and, secondly, that the intended models of such a theory are preserved by the OWL version.

The proofs, the documentation, and all the required files are available online[23]. In particular, the repository contains: *i*) the version of $\text{DOLCEsimple}_{\text{FOL}}$ and $\text{DOLCEsimpleNary}_{\text{FOL}}$ in the format of Prover9/Mace4 and the proof of their consistency; *ii*) the translation of $\text{DOLCEbasic}_{\text{OWL}}$ and $\text{DOLCEnaryRel}_{\text{OWL}}$ into first-order logic, in the format of Prover9 and Vampire (i.e., in the tptp format), *iii*) a report of the axioms of $\text{DOLCEbasic}_{\text{OWL}}$ and $\text{DOLCEnaryRel}_{\text{OWL}}$ proved from $\text{DOLCEsimple}_{\text{FOL}}$ and $\text{DOLCEsimpleNary}_{\text{FOL}}$ (respectively).

4.1 Adequacy of $\text{DOLCEbasic}_{\text{OWL}}$

We prove that $\text{DOLCEsimple}_{\text{FOL}}$ entails the axioms of $\text{DOLCEbasic}_{\text{OWL}}$. To achieve that, our strategy is to translate $\text{DOLCEbasic}_{\text{OWL}}$ into first-order logic, then use Prover9 to automatically prove each axiom of $\text{DOLCEbasic}_{\text{OWL}}$ from $\text{DOLCEsimple}_{\text{FOL}}$. The translation of $\text{DOLCEbasic}_{\text{OWL}}$ into first-order logic was done automatically by means of the translation tool developed

[21] Cf. `https://www.cs.unm.edu/~mccune/prover9/`
[22] Cf. `https://vprover.github.io`
[23] `https://github.com/appliedontolab/DOLCE/tree/main/OWL/Proof`

by [Flügel et al.2021].[24] A minor technicality is that DOLCEsimple$_{\mathrm{FOL}}$ uses the original abbreviated D18 labels of classes and relations (e.g., `ed` for `Endurant`, `p` for `partOf`), while DOLCEbasic$_{\mathrm{OWL}}$ employs fully spelled-out labels to comply with standard use. Thus, we translated the vocabulary of DOLCEbasic$_{\mathrm{OWL}}$ into the vocabulary of DOLCEsimple$_{\mathrm{FOL}}$.

Prover9 is able to directly prove each axiom of DOLCEbasic$_{\mathrm{OWL}}$ that does not involve the constant versions of the temporalised relations of DOLCE (e.g., `constantParticipantOf`), which are novel to DOLCEbasic$_{\mathrm{OWL}}$, as expected. To prove the axioms of DOLCEbasic$_{\mathrm{OWL}}$ that are about constant temporalised relations, we added their definitions to the (first-order) theory of DOLCEsimple$_{\mathrm{FOL}}$. Notice that such novel relations are definable in DOLCE, since they always use relations and classes that have already been axiomatised by the theory (e.g. mereology, participation, or constitution). For instance, the following object properties of DOLCEbasic$_{\mathrm{OWL}}$ are definable relations of DOLCEsimple$_{\mathrm{FOL}}$: `constantPartOf`, `constantProperPartOf`, `constantAatomicPartOf`. They are captured by means of axioms of the following form (omitting outermost universal quantifications).

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

$$\mathtt{constantProperPartof}(x,y) \leftrightarrow (\mathtt{ed}(x) \wedge \mathtt{ed}(y) \wedge \exists t\ \mathtt{pre}(y,t) \wedge \forall t(\mathtt{pre}(y,t) \rightarrow \mathtt{tpp}(x,y,t)))$$

$$\mathtt{constantAtomicPartof}(x,y) \leftrightarrow (\mathtt{ed}(x) \wedge \mathtt{ed}(y) \wedge \exists t\ \mathtt{pre}(y,t) \wedge \forall t(\mathtt{pre}(y,t) \rightarrow \mathtt{tatp}(x,y,t)))$$

where `ed` is the class `Endurant` and `pre`, `tp`, `tpp`, `tatp` are the relations being present, temporary parthood, temporary proper parthood, and temporary atomic parthood (respectively) defined in DOLCEsimple$_{\mathrm{FOL}}$.[25]

Notice that the definition of `constantlyOverlaps` departs from the previous pattern. It is motivated by the goal of proving the symmetry of `constantlyOverlaps` and the compatibility with `constantPartOf`, i.e., the theorem $\mathtt{constantPartOf}(x,y) \rightarrow \mathtt{constantlyOverlaps}(x,y)$.

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

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

[25] The relation of constant parthood was already present in D18, cf. Axiom Dd25.

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 discussed at the end of Section 3.3.[26]

The constant versions of the participation, constitution, and quale relations follow a similar pattern.

$\texttt{constantParticipantOf}(x, y) \leftrightarrow (\texttt{ed}(x) \wedge \texttt{pd}(y) \wedge \exists t\ \texttt{pre}(y, t) \wedge \forall t(\texttt{pre}(y, t) \rightarrow \texttt{pc}(x, y, t)))$

$\texttt{constantConstituentOf}(x, y) \leftrightarrow (((\texttt{ped}(x) \wedge \texttt{ped}(y)) \vee (\texttt{nped}(x) \wedge \texttt{nped}(y)) \vee (\texttt{pd}(x) \wedge \texttt{pd}(y))) \wedge \exists t\ \texttt{pre}(y, t) \wedge \forall t(\texttt{pre}(y, t) \rightarrow \texttt{k}(x, y, t)))$

$\texttt{constantQualeOf}(x, y) \leftrightarrow (((\texttt{ar}(x) \wedge \texttt{aq}(y)) \vee (\texttt{pr}(x) \wedge \texttt{pq}(y))) \wedge \forall t(\texttt{pre}(y, t) \rightarrow \texttt{tql}(x, y, t)))$

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

To conclude, the theory DOLCEsimple$_{\text{FOL}}$, augmented with the constant version of the temporalised relations can be proved consistent with mace4. Moreover, we were able to find models obtained by mace4 where the classes are populated by at least one instance. Finally, Prover9 can prove that DOLCEsimple$_{\text{FOL}}$ entails (the first-order translation of) each axiom of DOLCEbasic$_{\text{OWL}}$.

4.2 Adequacy of DOLCEnaryRel$_{\text{OWL}}$

The adequacy of DOLCEnaryRel$_{\text{OWL}}$ is less straightforward than the case of DOLCEbasic$_{\text{OWL}}$, since the original DOLCE did not introduce reified relationships. For this reason, we developed a first-order theory that includes DOLCEsimple$_{\text{FOL}}$ and extended it with the reifications of the required relationships. We term this theory DOLCEsimpleNary$_{\text{FOL}}$. As we discussed in Section 2.3, the reified relationships are separated from DOLCE particulars and no axiom of DOLCE applies to them. The first part of the extension provided by DOLCEsimpleNary$_{\text{FOL}}$ contains the axioms that define the taxonomy of relationships in first-order logic, following the analysis of Section 3.4. The second part of the theory is crucial to assess the viability of adding

[26]Notice that formula $\forall t \forall t'(\exists x \exists y(\texttt{temporallyLocatedAt}(x, t) \wedge \texttt{cov}(x, y) \wedge \texttt{temporallyLocatedAt}(y, t')) \rightarrow \texttt{ov}(t, t')))$ is a theorem of DOLCEsimple$_{\text{FOL}}$ and could be added as a suitable property chain to DOLCEbasic$_{\text{OWL}}$, but it clashes with Pellet in Protégé (ver. 5.6.3), so we omit it.

reified relationships to DOLCE. This part contains, for each type of reified relationships, a pattern of axioms that relate each reified relationship with the corresponding relational statement of DOLCE. We illustrate the idea for the case of the reification of temporal parthood.

(r_1) $\texttt{RRelTempPart}(r) \rightarrow (\exists xyt(\texttt{tp}(x, y, y) \wedge \texttt{tempPart}(r, x) \wedge \texttt{tempWhole}(r, y) \wedge \texttt{rRelTime}(r, t)))$

(r_2) $\texttt{tempPart}(x, y) \rightarrow ((\texttt{RRelTempPart}(x) \vee \texttt{RRelTempProperPart}(x)) \wedge \texttt{ed}(y))$

(r_3) $\texttt{tempWhole}(x, y) \rightarrow ((\texttt{RRelTempPart}(x) \vee \texttt{RRelTempProperPart}(x)) \wedge \texttt{ed}(y))$

(r_4) $\texttt{RRelTempPart}(x) \wedge \texttt{tempPart}(x, y) \wedge \texttt{tempPart}(x, z) \rightarrow y = z$

(r_5) $\texttt{RRelTempPart}(x) \wedge \texttt{tempWhole}(x, y) \wedge \texttt{tempWhole}(x, z) \rightarrow y = z$

Axiom (r_1) asserts that, whenever we have a relationship r which is a reification of a temporal parthood statement, there exist three particulars of DOLCE, x, y, and t, such that: *i*) $\texttt{tp}(x, y, t)$ holds, *ii*) the temporal part "argument" of r is x, *iii*) the temporal whole "argument" of r is y, and *iv*) the time "argument" of r is t. Axioms(r_2) and (r_3) define the types of the properties `tempPart` and `tempWhole` that relate the reified relationship with the arguments of the relational statement (the time argument is defined for all temporalised relationships by means of the property `rRelTime`), as seen on Figure 9 in Section 3.5. In this case, `tempPart` and `tempWhole` are also used to connect the arguments of the reifications of temporal proper parthood statements. Axioms (r_4) and (r_5) force the unicity of the arguments, for each reified relationships. The other reified relationships follow a similar pattern.

In our approach, we are assuming that, once a reified relationship is introduced in the theory, a corresponding statement of DOLCE must hold. This requirement is needed to show that the intended meaning of the reified relationships, from the perspective of $\text{DOLCEsimpleNary}_{\text{FOL}}$, is to assert that a certain formula of DOLCE holds. We are not imposing the other direction, namely, that every relational statement of DOLCE must be witnessed by a reification. This is motivated by the idea of viewing reifications as a technical move to improve OWL representation and not as an ontologically motivated extension of DOLCE. The only exception to this strategy is provided by the participation statements (pc(x,y,t)), see (r_7).

(r_6) $\texttt{RRelTempParticipation}(r) \rightarrow (\exists xyt(\texttt{pc}(x, y, y) \wedge \texttt{tempParticipationParticipant}(r, x)$

$\wedge$ tempParticipationPerdurant(r, y)
$\wedge$ rRelTime$(r, t)))$

(r_7) pc(x, y, t) $\rightarrow$ $(\exists r($RRelTempParticipation(r) $\wedge$ tempParticipationParticipant(r, x)
$\wedge$ tempParticipationPerdurant(r, y)
$\wedge$ rRelTime$(r, t)))$

Axiom (r_6) conforms to the previously discussed pattern. By contrast, (r_7) impose that, once a participation statement holds, a corresponding reified relationship exists. This exception is motivated because, without (r_7), DOLCEsimpleNary$_{\text{FOL}}$ cannot prove the axiom of DOLCEnaryRel$_{\text{OWL}}$ that states that every endurant must participate to a perdurant at a certain time, mirroring axiom Ad35 of D18.

We were able to prove that DOLCEsimpleNary$_{\text{FOL}}$ is consistent by using mace4 [27]. However, in this case, mace4 was not able to exhibit a model of DOLCEsimpleNary$_{\text{FOL}}$ where each class of the ontology, and in particular each class of reified relationships, is populated by at least one element. The reason is that, by adding all the required reified relationships, we are forcing the domain to contain many elements, exceeding the resources that mace4 could handle. To circumvent this problem, we crafted a (finite) model of DOLCEsimpleNary$_{\text{FOL}}$ and we tested that it verifies each axiom of DOLCEsimple$_{\text{FOL}}$. For the actual model and the details, we refer to our repository.[28] The strategy is briefly the following one. We start from a model (D, I) of DOLCEsimple$_{\text{FOL}}$ generated by mace4 and we populate the interpretation of each relation that corresponds to a reified relationship by at least one n-tuple of elements of the domain. Then, we add new distinct elements to the domain D to instantiate the leafs of the taxonomy of reified relationships (see Figure 7). Let (D, I) be a (finite) model of DOLCEsimple$_{\text{FOL}}$ where the interpretations of the following relations are non-empty: $\texttt{tatp}^I$ (temporal atomic part of), $\texttt{tp}^I$ (temporal part of), $\texttt{tpp}^I$ (temporal proper part of), $\texttt{tov}^I$ (temporal overlap), $\texttt{sumt}^I$ (temporal sum), $\texttt{csum}^I$ (constant sum), $\texttt{pc}^I$ (participation), $\texttt{k}^I$ (constitution), $\texttt{tql}^I$ (temporary quale of), $\texttt{sum}^I$ (sum). Let $\{p_1, \ldots, p_j, \ldots, p_k\}$ be an enumeration of all the triples $(a_j, b_j, t_j) \in \texttt{pc}^I$. We construct a model (D', J) of DOLCEsimpleNary$_{\text{FOL}}$ by setting $D' = D \cup \{d_1, \ldots, d_{13}, p_1 \ldots, p_k\}$, where for $i \neq j$ $d_i \neq d_j$ and $\{d_1, \ldots, d_{13}, p_1 \ldots, p_k\} \cap D = \emptyset$. That is, the elements from d_1 to d_{13} are used to populate the classes of relationships beside RRelTempParticipation,

[27] Cf. https://www.cs.unm.edu/~mccune/mace4/.
[28] Cf. https://github.com/appliedontolab/DOLCE

whereas $p_1, \ldots, p_k$ are dedicated to `RRelTempParticipation`. Moreover, we assume that J coincides with I on the signature of $\text{DOLCEsimple}_{\text{FOL}}$ (which is a subset of the signature of $\text{DOLCEsimpleNary}_{\text{FOL}}$).

We populate each leaf of the taxonomy of relationships with the new elements and interpret the other relations of the signature of $\text{DOLCEsimpleNary}_{\text{FOL}}$ accordingly. For instance, we illustrate the case of `RRelTempParticipation`.

$$\texttt{RRelTempParticipation}^J = \{p_1, \ldots, p_k\}$$

$$\texttt{tempParticipationParticipant}^J = \{(p_1, a_1), \ldots, (p_k, a_k)\}$$

$$\texttt{tempParticipationPerdurant}^J = \{(p_1, b_1), \ldots, (p_k, b_k)\}$$

$$\texttt{rRelTime}^J = \{(p_1, t_1), \ldots, (p_k, t_k)\}.$$

This model verifies axioms of the type (r_6) and (r_7) by construction. Axioms of type (r_2) and (r_3) rephrased for `rRelTempParticipation` are verified because $\texttt{pc}(a_j, b_j, t_j)$ entails in $\text{DOLCEsimple}_{\text{FOL}}$ that a_j, b_j and t_j are of the right types. Axioms of types (r_4) and (r_5) are verified by construction of J.

In this case, Prover9 was not able to prove that $\text{DOLCEsimpleNary}_{\text{FOL}}$ entails (the first-order translation of) each axiom of $\text{DOLCEnaryRel}_{\text{OWL}}$, whereas Vampire succeeded. Therefore, every model of DOLCE extended with the reified relationships is preserved by our OWL 2 rendering.

5 Use Cases

We present in the following subsections two case studies with the goal of illustrating how to employ the OWL classes and properties of $\text{DOLCEbasic}_{\text{OWL}}$ and $\text{DOLCEnaryRel}_{\text{OWL}}$. Both examples are based on the case studies presented by [Borgo et al.2022].[29] Table 2 lists the namespace prefix bindings for the two use cases, in addition to the definitions in Table 1.

5.1 Case on Composition and Constitution

The first use case, which is based on the Case 1 presented by [Borgo et al.2022], shows the modeling of the composition and constitu-

[29]The logical consistency and reasoning results documented in the case studies were tested with the Pellet reasoner. Our tests contained dozens of instances and the reasoner answered almost instantaneously.

Table 2: List of namespaces with prefix names of the use cases.

Prefix	**IRI**	**Use case**
c01:	`https://w3id.org/DOLCE/OWL/UC/CompConst01#`	Sect.5.1
q01:	`https://w3id.org/DOLCE/OWL/UC/Quality01#`	Sect.5.2

tion 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 across time by its tabletop component, an essential part. Also, the existence of the table does not imply that it is made of the same matter throughout its whole life; thus, if one of the components of the table changes, the amount of matter in the whole table changes, too.

The table undergoes three lifecycle phases:

P1 During the time interval `t0`, following manufacturing, a wooden table (`T`) consists of a table top (`Ttop`) and four legs (`Leg1, Leg2, Leg3, Leg4`). The tabletop and legs are made from amounts of wood (`Wtop, W1, W2, W3, W4`, respectively).

P2 During the time interval `t1`, following maintenance, component `Leg4` is replaced with the new leg `Leg4new`, which is made from the amount of matter `W4new`.

P3 During the time interval `t2`, after the table is dismantled, the table ceases to exist, but the amounts of wood of its components remain intact.

To represent this case study, we need to take into account at least three aspects of the considered domain entities, namely, that the table consists of different components, that the table and its components are constituted by amounts of matter, and that all the entities are present at certain time intervals. From a modeling perspective, we therefore need to rely on the use of the relations of proper parthood, constitution, and presence. As we have seen in the previous sections, according to DOLCEbasic$_{\mathrm{OWL}}$ and DOLCEnaryRel$_{\mathrm{OWL}}$, the first two relations can be considered as either *constant* (OWL object properties) or *reified* relationships (OWL classes), see Table 3. From a methodological standpoint, using one or the other approach requires making some choices from the modeler's side. For instance, reified relationships are suited for application contexts where the focus is on the possibility that domain entities undergo changes, differently from constant relations which represent conditions holding for the entire life of the entities. As we will show in the following, one may also take a *hybrid approach*, where some conditions are represented in the constant way, and others in the reified manner.

Table 3: Options for constant and temporalised relationships of parthood and constitution.

Relation	**Object property**	**Class**
Parthood	`db:constantProperPartOf`	`dn:RRelTempProperPart`
Constitution	`db:constantConstituentOf`	`dn:RRelTempConstitutionPED`

To instantiate this first case study, we assume that the tabletop `Ttop` is a constant part of the table. On the other hand, we represent the table legs `Leg1` ... `Leg4new` as parts that may change over time. Additionally, we assume that each table component is constantly constituted by its amount of matter. Taking these factors into account, the object properties for the case study are as follows:

- `constantProperPartOf`: to specify the *constant* components of the table, namely, the tabletop.
- `constantConstituentOf`: to specify the *constant* amounts of matter of the table's components.
- `presentAt`: to specify the presence in time of domain entities.

The diachronic representation of the table legs requires the instantiation of the temporalised reified relationship `RRelTempProperPart`. As seen in Section 3.4, this involves the use of the following object properties to specify its arguments:

- `tempWhole`: the whole taking part in the reified relationship, e.g. the table;
- `tempPart`: the part of the whole, e.g. a specific table leg;
- `rRelTime`: the time interval during which the relationship between the whole and the part holds.

The domain entities for the use case belong to the following classes:

- `TimeInterval` from DOLCEbasic$_{\mathrm{OWL}}$;

- Artefact, i.e. a subclass of NonAgentivePhysicalObject, introduced for the sake of this use case according to [Borgo et al.2022];
- Table, TableTop, and TableLeg defined as novel subclasses of Artefact;
- AmountOfWood, i.e. a novel subclass of AmountOfMatter for this use case.

Figure 14 shows the novel classes specializing DOLCEbasic$_{\mathrm{OWL}}$ and the related instances. In addition, individuals RRTpp01 ... RRTpp08 instantiate the reified relationship class RRelTempProperPart. Readers can browse the representation of the case study in the available OWL file.[30]

The constant parthood and constitution relations can be exemplified by looking at Ttop and Leg4/Leg4new components in Figures 16 and 15, respectively.

Considering Figure 15, the amount of wood Wtop is a constant constituent of table top Ttop, which is a constant proper part of table T. The table is present at time intervals t0 and t1. The presence in time (db:presentAt) of several individuals is automatically inferred by reasoning over the Abox of the case study and the axioms of the ontologies. The presence of Ttop at t0 and t1 is inferred by reasoning over constant parthood: a part (Ttop) is present whenever its whole (T) is present. Then, the presence of Wtop at t0 and t1 is inferred by reasoning over constant constitution: the material (Wtop) is present whenever the constituted individual (Ttop) is present.

Looking now at Figure 16, Leg4 and Leg4new are represented as temporary components of table T by instantiating the reified relationship RRelTempProperPart with the individuals RRTpp07 and RRTpp08. Notice that also in this case various cases of presence in time can be automatically computed by reasoning. In particular, the formal characterization of reified temporalised relationships with property chains (see Section 3.6) leads to infer that the arguments of the relationship are present when the relationship holds. Therefore, it is inferred that Leg4 is present at t0 (because Leg4 is an argument of RRTpp07 that has time parameter t0) and Leg4new is present at t1 (because Leg4new is an argument of RRTpp08 that has time parameter t1). The presence of table at time intervals t0 and t1 has been explicitly stated; however, it could also be inferred similarly.

In addition, by reasoning over constant constitution one infers that W4 is present at t0 and W4new is present at t1.

Finally, Figures 15 and 16 make explicit the presence of both W4new and Wtop at t2. This is because, as said, according to the case study the amounts

[30] https://w3id.org/DOLCE/OWL/UC/CompConst01

of matter continue to exist in time even when the table is destroyed, which is meant to occur during `t2`.

5.2 Case on Qualities

This second use-case is based on Case 3 presented by [Borgo et al.2022] and is relative to the modeling of individual qualities of endurants. For the sake of the discussion, we consider the location in space of a building and the color of its facade in order to show, as for the previous case, the modeling of qualities in both constant and temporalised ways.

From a general perspective, recall that DOLCE distinguishes between physical, abstract, and temporal qualities depending on their inherence in physical endurants, non-physical endurants or perdurants, respectively (see Section 3.2). Also, the value (called quale in the theory) of a quality is represented as a region in a (abstract, physical, temporal) quality space (see Fig. 6). Since we deal in our case study with a physical endurant like a building, we shall focus here on the representation of physical qualities and their values.

The object property `directQualityOf` in DOLCEbasic$_{\mathrm{OWL}}$ is the relation used to relate a (abstract, physical, temporal) quality to the unique particular it inheres in. To represent the values of the qualities of endurants, either a constant or temporalised approach can be adopted (see Table 4). In particular:

- `constantQualeOf` from DOLCEbasic$_{\mathrm{OWL}}$: relates a (abstract or physical) quality to its *constant* value. Being a constant relation, its intended meaning is that x is the value of quality y during the *whole* existence of y.

- `RRelTempQuale` from DOLCEnaryRel$_{\mathrm{OWL}}$: models the relationship between a (physical or abstract) quality, a region, and a time (interval) at which the quality has this region as value. As can be seen in Figure 12, this reified relationship is specialized into `RRelTempQualePRPQ` for physical qualities, and `RRelTempQualeARAQ` for abstract qualities.

The following object properties relate the relationships in `RRelTempQuale` and its subclasses to their three arguments (see Section 3.4):

- `tempQualeQuality`: represents the quality;

- `tempQualeRegion`: represents the value of the quality;

- `rRelTime`: represents the time at which the reified relationships between the quality and its value holds.

Table 4: Options for the constant relation and reified relationships of *quale* (position in a quality space)

Relation	**Object property**	**Class**
Quale	`db:constantQualeOf`	`dn:RRelTempQuale`, `dn:RRelTempQualePRPQ`, `dn:RRelTempQualeARAQ`

As for the case study presented in the previous section, when addressing the representation of qualities, modellers must decide which approach to use depending on their requirements. In our example, it is reasonable to model the physical location of the building as being constant for the entire lifespan of the building, while the color of its facade can undergo changes. For instance, let us assume that it is yellow during the time interval t_1, and green during t_2.

The domain entities for the use case belong to the following classes:

- `TimeInterval` from DOLCEbasic$_{\text{OWL}}$;
- `Bulding` and `Facade`, i.e. two novel subclasses of `NonAgentivePhysicalObject` (a subclass of `PhysicalEndurant`) from DOLCEbasic$_{\text{OWL}}$;
- `ColorQuality`, i.e. a novel subclass of `PhysicalQuality`;
- `SpatialLocation`, i.e. a native subclass of `PhysicalQuality` for representing the location in space of physical endurants;
- `ColorRegion`, i.e. a novel subclass of `PhysicalRegion` for representing the values of color qualities;
- `SpaceRegion`, i.e. a native subclass of `PhysicalRegion` for representing the position in space of physical endurants.

The novel classes specialising DOLCEbasic$_{\text{OWL}}$ and the related instances are depicted in Figure 17. In addition, individuals `RRTqT1` and `RRTqT2` instantiate the reified relationship class `RRelTempQualePRPQ`. Readers can browse the representation of the case study in the available OWL file.[31]

Figure 18 shows that the spatial location `spatialLoc01` is quality of the building `building01` with constant quale `placeXYZ`, this standing for the location in space of the building. The building is present at time interval `t` that has proper parts `t1` and `t2`, therefore it is inferred that the building is present at `t1` and `t2` as well. In addition, the presence of the building spatial location can be inferred.

Looking at Figure 19, `facade01` is a constant proper part of `building01` and has `colorQuality01` as its individual color-quality. The quale (value) of this quality changes over time. This is represented by instantiating the reified relationship `RRelTempQualePRPQ` with `RRTqT1` and `RRTqT2`. Specifically, `RRTqT1` represents the position of `colorQuality01` in the region `colorYellow` at `t1`, whereas `RRTqT2` represents the positions of the quality in `colorGreen` at `t2`. The presence of the facade at `t`, `t1`, `t2` is inferred from the proper parthood relation with the building that is present at these time intervals. The presence of the color-quality is inferred from the presence of the facade.

5.3 Remarks

Commenting on the examples, as a remark relative to the representation of qualities, in the case of the spatial location of a physical endurant like a building, one may wish to represent coordinates in space, e.g. by reusing geospatial data. Similarly, one may wish to represent the quantitative value of, say, the dimensions of an endurant along with the associated measurement system, e.g., a numerical value of weight in kilos. For these purposes, one could explore and integrate existing work janowicz2019,rijgersberg2013. Future work will address the introduction of modelling patterns for representing quantitative qualities values by exploiting the full capabilities of OWL 2, including the use of data properties.

With respect to the use of reified relationships, as said, the use of time indexes (with object property `dn:rRelTime`) conveys information relative to the time when the relationship holds. For instance, `RRTqT1` in the second case study holds at time $t1$. It is important to stress that the time does not model the presence of the reified relationship. This because, as explained in Sect. 2.3, reified relationships are introduced for technical purposes only; they do not carry an ontological meaning, and are not particulars of DOLCE, thus they are not in time. Also, the information conveyed by the reified relationship holds only for the time specified through `dn:rRelTime`. For instance, considering Fig. 19, `RRTqT1` holds only at time $t1$. If the time index

[31] `https://w3id.org/DOLCE/OWL/UC/Quality01`

t of a reified relationship with arguments x and y is a non-atomic interval, there should exist further reified relationships at each atomic instant within t, having the same arguments x and y. This cannot be however enforced in OWL because of its restrictive expressivity.

6 Conclusions and further work

We presented the core aspects of our approach to DOLCE based on OWL 2. In particular, we designed two modules: (*i*) the $\text{DOLCEbasic}_{\text{OWL}}$ module, that provides the taxonomy of DOLCE, its main binary relations, and the constant versions of the temporalised relations and (*ii*) the $\text{DOLCEnaryRel}_{\text{OWL}}$ module, that approaches the case of n-ary relations of DOLCE, for $n > 2$. Unlike other existing Semantic Web formal representations of DOLCE, we placed primary importance on adhering to the original D18 spirit of the ontology. During the development of our OWL 2 rendering, delicate ontological issues have arisen, such as properly understanding the constant version of the temporalised relations of DOLCE and the viable introduction of reified relationships into DOLCE. Moreover, we have established our adequacy condition by proving that the intended models of DOLCE and DOLCE added with reified relationships are preserved by our OWL 2 ontologies. Finally, we tested our ontologies by means of two use cases to demonstrate their functionality and provide guidance on how to use them. Further testing is a matter for future work. In particular, we plan to test the performance of our modules with a large number of assertions and to evaluate their scalability and efficiency under realistic workloads.

We view $\text{DOLCEbasic}_{\text{OWL}}$ and $\text{DOLCEnaryRel}_{\text{OWL}}$ as the first modules in a rich library of DOLCE-based modules. In compliance with DOLCE-driven research, another module will comprise the representation of *concepts* and *descriptions* masolo2004social, which often find their place for representing (social) roles but also engineering technical specifications, as in the work of [Terkaj et al.2022], among other entities. A further module will introduce the use of OWL 2 data properties, especially for enriching the representation of qualities and qualia when the latter are characterized through numerical values DBLP:conf/ki/PorelloRT0K23. Last but not least, current research aims at developing modules for specific application domains, including product design and manufacturing terkaj2022ontology.

From an ontology design perspective, the development of the DOLCE-modules library will take advantage of existing Semantic Web resources, such as the W3C Time Ontology[32] and SSN/SOSA.[33] At least a partial integration of these resources is a matter for future work.

Acknowledgments: We would like to thank our colleagues at the CNR ISTC Laboratory for Applied Ontology for their valuable feedback and discussions, which led to the OWL version of DOLCE presented in this paper. The authors of this paper are solely responsible for any remaining errors.

References

[Benevides et al.2019] Benevides, A.B., Bourguet, J., Guizzardi, G., Peñaloza, R. & Almeida, J.P.A. (2019). Representing a reference foundational ontology of events in SROIQ. *Appl. Ontology*, 14(3), 293–334. doi:10.3233/AO-190214.

[Borgo et al.2022] Borgo, S., Ferrario, R., Gangemi, A., Guarino, N., Masolo, C., Porello, D., Sanfilippo, E.M. & Vieu, L. (2022). DOLCE: A Descriptive Ontology for Linguistic and Cognitive Engineering. *Applied Ontology*, 17(1), 45–69. doi:10.3233/AO-210259.

[Flügel et al.2021] Flügel, S., Kleinau, A., Neuhaus, F., Glauer, M. & Hastings, J. (2021). 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. CEUR Workshop Proceedings*. CEUR-WS.org. `https://ceur-ws.org/Vol-2969/paper7-DEMO.pdf`.

[Garbacz2022] Garbacz, P. (2022). 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. CEUR Workshop Proceedings*. CEUR-WS.org. `https://ceur-ws.org/Vol-3249/paper1-FOUST.pdf`.

[Hahmann and Powell II2021] Hahmann, T. & Powell II, R.W. (2021). Automatically Extracting OWL Versions of FOL Ontologies. In A. Hotho,

[32] `https://www.w3.org/TR/owl-time/`.
[33] `https://www.w3.org/TR/vocab-ssn/`.

E. Blomqvist, S. Dietze, A. Fokoue, Y. Ding, P.M. Barnaghi, A. Haller, M. Dragoni and H. Alani (Eds.), *The Semantic Web - ISWC 2021 - 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24-28, 2021, Proceedings. Lecture Notes in Computer Science* (Vol. *12922*, pp. 252–269). Springer. doi:10.1007/978-3-030-88361-4_15.

[Horrocks and Sattler2004] Horrocks, I. & Sattler, U. (2004). Decidability of $\mathcal{SHIQ}$ with complex role inclusion axioms. *Artificial Intelligence*, 160(1–2), 79–104. doi:10.1016/j.artint.2004.06.002.

[Kutz and Mossakowski2011] Kutz, O. & Mossakowski, T. (2011). A Modular Consistency Proof for DOLCE. In W. Burgard and 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. http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3754.

[Ligozat1991] Ligozat, G. (1991). On Generalized Interval Calculi. In *AAAI Conference on Artificial Intelligence.*

[Porello et al.2024] Porello, D., Vieu, L., Terkaj, W., Borgo, S., Compagno, F. & Sanfilippo, E.M. (2024). DOLCE in OWL: The Core Theory. In I. Oliveira, P.P.F. Barcelos, R. Calhau, C.M. Fonseca and G. Righetti (Eds.), *Proceedings of the Joint Ontology Workshops 2024, EPISODE X: THE TUKKER ZOMER OF ONTOLOGY. CEUR Workshop Proceedings* (Vol. *3882*, pp. 1–14). CEUR-WS.org. https://ceur-ws.org/Vol-3882/foust-1.pdf.

[Terkaj et al.2022] Terkaj, W., Borgo, S. & Sanfilippo, E.M. (2022). Ontology for Industrial Engineering: A DOLCE Compliant Approach. In *CEUR Workshop Proceedings* (Vol. *3240*).

A OWL entities

Table 5: List of classes in DOLCEbasic$_{\text{OWL}}$. Prefix db: is omitted for all classes.

Local IRI	**Label**	**D18 Acronym**	**Comment**
Abstract	Abstract	AB	An Abstract Entity is an entity that has neither spatial nor temporal qualities and is not a quality itself (examples: a number, a set, a quality space).
AbstractQuality	Abstract Quality	AQ	An Abstract Quality is a quality that directly inheres to a non-physical endurant (examples: the value of an asset, the rights of the UN general secretary).
AbstractRegion	Abstract Region	AR	An Abstract Region is a region in the abstract quality space for abstract qualities (example: the conventional value of 1 Euro).
Accomplishment	Accomplishment	ACC	An Accomplishment is an event which is mereologically non-atomic (examples: a conference, an ascent, a performance).
Achievement	Achievement	ACH	An Achievement is an event which is mereologically atomic (examples: reaching the summit of K2, a departure, a death).
AgentivePhysicalObject	Agentive Physical Object	APO	An Agentive Physical Object is a physical object to which intentions, believes and desires are ascribed (examples: a human person as opposed to a legal person).
AgentiveSocialObject	Agentive Social Object	ASO	An Agentive Social Object is a social object to which intentions, believes and desires are ascribed (examples: a teacher, a president, a software agent, a nation).
AmountOfMatter	Amount of Matter	M	An Amount Of Matter is a physical endurant which has no unity criterion and is mereologically invariant, that is, all its part are essential parts (examples: the gold of my wedding ring, the sand used to make this glass).
ArbitrarySum	Arbitrary Sum	ASO	An Arbitrary Sum is an endurant which is the mereological sum of at least a physical and a non-physical endurant.
Atom	Atom	At	Atom (atomic Abstract or Perdurant)
ConstantAtom	Constant Atom	n/a	A constant atom is an endurant that has no constant proper parts
Endurant	Endurant	ED	An Endurant is an entity wholly present, i.e., all its proper parts are present, at any time that it is present (examples: a person, a tree, an atom, an idea).
Event	Event	EV	An Event is an anti-cumulative perdurant, i.e., a perdurant that when summed to another perdurant of the same type, gives a perdurant of a different type (examples: closing a door, reaching the top of a mountain, breaking a seal).
Feature	Feature	F	A Feature is a physical endurant which has a unity criterion and is existentially dependent on another physical endurant, its host (examples: a hole, a bump, an object's boundary, a stain on a t-shirt).
MentalObject	Mental Object	MOB	A Mental Object is a non-physical object which existentially depends on an agentive physical object (examples: a percept, a sense datum).
NonAgentivePhysicalObject	Non-Agentive Physical Object	NAPO	A Non-Agentive Physical Object is a physical object to which intentions, believes and desires are not ascribed (examples: a pebble, a house, a computer, a human body).

Table 5: List of classes in DOLCEbasic$_{\mathrm{OWL}}$ (continued)

Local IRI	**Label**	**D18 Acronym**	**Comment**
NonAgentiveSocialObject	Non-Agentive Social Object	NASO	A Non-Agentive Social Object is a social object to which intentions, believes and desires are not ascribed (examples: a law, an economic system, a currency, an asset).
NonPhysicalEndurant	Non-Physical Endurant	NPED	A Non-Physical Endurant is an endurant with no direct spatial quality (examples: democracy, the United Nations, the general secretary of Amnesty International).
NonPhysicalObject	Non-Physical Object	NPOB	A Non-Physical Object is a non-physical endurant with unity criterion (examples: a theory, a topic, a concept).
Particular	Particular	PT	A Particular is an entity that cannot have instances, in this sense it is opposed to universals (examples: a person, a soccer game, a plan, the color red, a feeling, a fact, a theory, number forty-two).
Perdurant	Perdurant	PD	A Perdurant is an entity that happens in time (examples: a person's life, a soccer game, the reading of a book, the state of feeling happy). Some perdurants are temporally atomic (example: the change of a letter's state due to the breaking of the seal).
PhysicalEndurant	Physical Endurant	PED	A Physical Endurant is an endurant with direct spatial quality (examples: a person, a tree, an atom, the water in a glass, a hole in a wall, the center of the Earth).
PhysicalObject	Physical Object	POB	A Physical Object is a physical endurant with unity criterion (examples: a person, a human body, a house, a computer).
PhysicalQuality	Physical Quality	PQ	A Physical Quality is a quality that directly inheres to a physical endurant (examples: the weight of a pen, the color of an apple).
PhysicalRegion	Physical Region	PR	A Physical Region is a region in a quality space for physical qualities (examples: the physical space, an area in the color quality space, 80Kg).
Process	Process	PRO	A Process is a stative perdurant such that at some temporal scale some parts are of a different type (examples: running, writing).
Quality	Quality	Q	A Quality is an entity that inheres in an endurant or a perdurant, and that can be perceived or measured (examples: the shape of a book, the color of a car, the size of a t-shirt, the electrical charge of a battery).
Region	Region	R	A Region is an abstract entity classified by a quality type and with mereological structure, the sum of all the regions of a quality type is called a quality space (examples: the color red is a region in the quality space of color, the red of a rose is a (sub)region of the red region in the quality space of color, the commercial value of 1 Euro is a region in the quality space of commercial values).
SocialAgent	Social Agent	SAG	A Social Agent is an individual agentive social object (examples: a teacher, a president, a software agent).
SocialObject	Social Object	SOB	A Social Object is a non-physical object which is generically dependent on a community of agents (examples: a person in the legal sense, a client, a law, an economic system).
Society	Society	SC	A Society is a collective agentive social object (examples: a nation, the FCA Group, Apple, the European Central Bank).
SpaceRegion	Space Region	S	A Space Region is a physical region, possibly disconnected, for spatial locations (examples: the region occupied by Earth in this moment, the region where it snowed in the year 1900).

Table 5: List of classes in $\text{DOLCEbasic}_{\text{OWL}}$ (continued)

Local IRI	**Label**	**D18 Acronym**	**Comment**
SpatialLocation	Spatial Location	SL	A Spatial Location is the individual spatial quality of a physical endurant (examples: the spatial quality of the Earth, the spatial quality of a hole, the spatial quality of Adolf Anderssen).
State	State	ST	A State is a stative perdurant such that all its parts are of the same type (examples: being sitting, being open, being happy, being red).
Stative	Stative	STV	A Stative is a cumulative perdurant, i.e., a perdurant that when summed to another perdurant of the same type, gives a perdurant of the same type (examples: sitting, walking, waiting).
TemporalLocation	Temporal Location	TL	A Temporal Location is a temporal quality of a perdurant and is specific to that perdurant, that is, two concurrent perdurants have distinct temporal locations. The temporal location is the temporal quality which has as quale the unique time interval at which the perdurant is exactly located (examples: the temporal location of World War I, the temporal location of the Anderssen-Kieseritzky chess play in 1851, the temporal location of the summers of the 20th century).
TemporalQuality	Temporal Quality	TQ	A Temporal Quality is a quality that directly inheres to a perdurant (example: the duration of World War I, the starting time of the 2000 Olympics).
TemporalRegion	Temporal Region	TR	A Temporal Region is a region in a quality space for temporal qualities (examples: a time interval, a historical period, a velocity).
TimeInterval	Time Interval	T	A Time Interval is a temporal region which is an interval or a sum of intervals. A Time Interval can be atomic, i.e., instantaneous. (examples: the first second of your life, the Cretaceous time period, the time interval of all winters of the third millennium).

Table 6: List of object properties in $\text{DOLCEbasic}_{\text{OWL}}$. Prefix `db:` is omitted for all classes.

Local IRI	**Label**	**D18 Acronym**	**Comment**
atomicPartOf	atomic part of	AtP	atomicPartOf is the specialization of the PartOf relation to the case that the part argument is a mereological atom
constantAtomicPartOf	constant atomic part of	n/a	constantAtomicPartOf is the specialization of the ConstantPartOf relation to the case that the part argument is a mereological atom
constantConstituentOf	constant constituent of	SK	constantConstituentOf is the constant version of Constitution which is a ternary relation between two endurants x, y and time interval t or between two perdurants x, y and time interval t. The intended interpretation is that x constitutes y over the time interval t. In this constant version, the entity x constitutes y during the whole existence of y.

Table 6: List of object properties in $\text{DOLCEnaryRel}_{\text{OWL}}$ (continued)

Local IRI	**Label**	**D18 Acronym**	**Comment**
constantPartOf	constant part of	CP	constantPartOf is the constant version of Temporary Parthood which is a ternary relation between two endurants x, y and a time interval t. The intended interpretation is that the endurant x is a part of the endurant y over the time interval t. In this constant version, the endurant x is part of the endurant y during the whole existence of y.
constantParticipantOf	constant participation of	PC_C	constantParticipantOf is the constant version of Participation which is a ternary relation between an endurant x, a perdurant y, and a time interval t. The intended interpretation is that x participates in y at time t. Every perdurant has at least an endurant that participates in it, and every endurant participates in some perdurant. In this constant version, the endurant x participates in the perdurant y during the whole existence of y.
constantProperPartOf	constant proper part of	n/a	constantProperPartOf is the irreflexive specialization of the constantPartOf relation.
constantQualeOf	constant quale of	n/a	constantQualeOf is the constant version of Temporary Quale Of which is a ternary relation among a physical region x, a physical quality y and a time interval t or among an abstract region x, an abstract quality y and a time interval t. It relates the quality to its quale (position) at a certain time. In this constant version, the region x is the quale of the quality y during the whole existence of y.
constantlyOverlaps	constantly overlaps	n/a	constantlyOverlaps holds between two endurants that share a constant part, and are such that the time intervals at which they are temporally located overlap.
directQualityOf	direct quality of	dqt	directQualityOf is a binary relation between a quality and the unique entity it inheres in, which is an endurant or a perdurant. It implies specific constant dependence.
overlaps	overlaps	O (binary)	overlaps holds between two abstracts or two perdurants when they both share a part.
partOf	part of	P (binary)	PartOf is a binary relation that is a partial ordering between abstract entities or between perdurants. It founds an extensional mereology on each of these two categories.
presentAt	present at	PRE	presentAt is a binary relation between an endurant or a perdurant or a quality x and a time interval t which is part of x's temporal location.
properPartOf	proper part of	PP (binary)	properPartOf is the irreflexive specialization of the partOf relation. (Irreflexivity not captured here, as transitivity is favored)
qualeOf	quale of	ql (binary)	qualeOf is a binary relation between a temporal region and a temporal quality, relating the quality to its quale (position in the quality space).
specificallyDependsOn	specifically depends on	SD	specificallyDependsOn is a binary relation between two entities x, y present in time, i.e., endurants, perdurants or qualities, with the intended interpretation that the existence of x specifically and constantly depends on the existence of y.

Table 6: List of object properties in $\text{DOLCEnaryRel}_{\text{OWL}}$ (continued)

Local IRI	**Label**	**D18 Acronym**	**Comment**
temporalPartOf	temporal part of	P (binary)	temporalPartOf is the specialisation of the partOf relation between two perdurants x and y, x being a temporal slice of y, that is, a maximal part of y during some time interval.
temporallyLocatedAt	temporally located at	ql_T	temporallyLocatedAt is a binary relation between an endurant or a perdurant or a quality and the unique time interval it is exactly located in.

Table 7: List of classes in $\text{DOLCEnaryRel}_{\text{OWL}}$. Prefix `dn:` is omitted for all classes.

Local IRI	**Label**	**Related to D18 Symbol**	**Comment**
RRelConstantSum	Reified relationship for constant sum	n/a	identifier for a constant sum relation
RRelSum	Reified relationship for sum	+	identifier for a sum relation
RRelSumAB	Reified relationship for sum of abstract	+	identifier for a sum relation restricted to abstract (AB)
RRelSumPD	Reified relationship for sum of perdurant	+	identifier for a sum relation restricted to perdurant (PD)
RRelTempAtomicPart	Reified temporalised relationship for atomic part	AtP (ternary)	To enable more inferences, you could add a subclass axiom such that RRelTempAtomicPart is a subclass of RRelTempProperPart. We do not include it here to avoid committment on the ontological nature of relation instances. If you do add this axioms, don't forget to add the corresponding inclusion of object properties.
RRelTempConstitution	Reified temporalised relationship for constitution	K	identifier for a constitution relation
RRelTempConstitutionNPED	Reified temporalised relationship for constitution of non-physical endurant	K	identifier for a constitution relation restricted to non-physical endurant (NPED)
RRelTempConstitutionPD	Reified temporalised relationship for constitution of perdurant	K	identifier for a constitution relation restricted to perdurant (PD)
RRelTempConstitutionPED	Reified temporalised relationship for constitution of physical endurant	K	identifier for a constitution relation restricted to physical endurant (PED)
RRelTempOverlap	Reified temporalised relationship for overlap	O (ternary)	identifier for a ternary overlap relation

Table 7: List of classes in DOLCEnaryRel$_{\text{OWL}}$ (continued)

Local IRI	**Label**	**Related to D18 Symbol**	**Comment**
RRelTempPart	Reified temporalised relationship for part	P (ternary)	To enable more inferences, you could add a subclass axiom such that RRelTempPart is a subclass of RRelTempOverlap. We do not include it here to avoid committment on the ontological nature of relation instances. If you do add this axioms, don't forget to add the corresponding inclusion of object properties (e.g. tempPart subproperty of tempOverlapArg1 etc.)
RRelTempParticipation	Reified temporalised relationship for participation	PC	identifier for a participation relation
RRelTempProperPart	Reified temporalised relationship for proper part	PP (ternary)	To enable more inferences, you could add a subclass axiom such that RRelTempProperPart is a subclass of RRelTempPart. We do not include it here to avoid committment on the ontological nature of relation instances.
RRelTempQuale	Reified temporalised relationship for quale	ql (ternary)	Identifier for a quale relation
RRelTempQualeARAQ	Reified temporalised relationship for quale of abstract region and abstract quality	ql (ternary)	Identifier for a quale relation restricted to abstract region (AR) and abstract quality (AQ).
RRelTempQualePRPQ	Reified temporalised relationship for quale of physical region and physical quality	ql (ternary)	Identifier for a quale relation restricted to physical region (PR) and physical quality (PQ).
RRelTempSum	Reified temporalised relationship for sum	$+_{te}$	Identifier for a temporary sum relation
ReifiedRelationship	Reified relationship	n/a	Identifier of an N-ary reified relationship. This class is not defined in DOLCE, but it is introduced because OWL is limited to binary relations.
ReifiedTemporalisedRelationship	Reified temporalised relationship	n/a	Identifier of a generic time-dependant N-ary relation instance. This class is not defined in DOLCE, but it is introduced to group temporalised relationships.

Table 8: List of object properties in DOLCEnaryRel$_{\text{OWL}}$. Prefix `dn:` is omitted for all object properties.

Local IRI	**Label**	**Related to D18 Symbol**	**Comment**
atomAt	temporary atom	AtP (ternary)	x is atomAt t if x is a mereologica atom part of some whole at time t

Table 8: List of object properties in $\text{DOLCEnaryRel}_{\text{OWL}}$ (continued)

Local IRI	**Label**	**D18 Acronym**	**Comment**
constantSumAddend1	first addend argument in constant sum relation	n/a	second argument of the constant sum relation
constantSumAddend2	second addend argument in constant sum relation	n/a	third argument of the constant sum relation
constantSumResult	sum argument of constant sum relation	n/a	first argument of the constant sum relation, this is the sum of the second and the third argument
rRelTime	time index of time-dependent relation	K, P (ter.), PC, ql (ter.), $+_{te}$, O (ter.), AtP (ter.)	Time index of all time-dependant relationships.
sumAddend1	first addend argument in sum relation	+	second argument of the sum relation
sumAddend2	second addend argument in sum relation	+	third argument of the sum relation
sumResult	sum argument in sum relation	+	first argument of the sum relation, this is the sum of the second and the third argument
tempAtomicPart	part argument in temporary atomic part relationship	AtP (ternary)	first argument of the temporary atomic part relation (RRelTempAtomicPart)
tempAtomicWhole	whole argument in temporary atomic part relationship	AtP (ternary)	second argument of the temporary atomic part relation (RRelTempAtomicPart)
tempConstitutionConstituent	constituent argument in constitution relation	K	first argument of the constitution relation
tempConstitutionConstituted	constituted argument in constitution relation	K	second argument of the constitution relation
tempOverlapArg1	first argument in temporary overlap relationship	O (ternary)	first argument of the temporary overlap relation
tempOverlapArg2	second argument in temporary overlap relationship	O (ternary)	second argument of the temporary overlap relation
tempPart	part argument in temporary part relationship	P (ternary)	first argument of the temporary part relationship defining the part
tempParticipationParticipant	participant argument in participation relationship	PC	first argument of the participation relation
tempParticipationPerdurant	perdurant argument in participation relation	PC	second argument of the participation relation

Table 8: List of object properties in $\text{DOLCEnaryRel}_{\text{OWL}}$ (continued)

Local IRI	**Label**	**D18 Acronym**	**Comment**
tempQualeQuality	quality argument in temporary quale relation	ql (ternary)	second argument of the temporary quale relation
tempQualeRegion	region argument in temporary quale relation	ql (ternary)	first argument of the temporary quale relation
tempSumAddend1	first addend argument in temporary sum relation	$+_{te}$	second argument of the temporary sum relation
tempSumAddend2	second addend argument in temporary sum relation	$+_{te}$	third argument of the temporary sum relation
tempSumResult	sum argument in temporary sum relation	$+_{te}$	first argument of the temporary sum relation, this is the sum of the second and the third argument
tempWhole	whole argument in temporary part relationship	P (ternary)	second argument of the temporary part relationship defining the whole

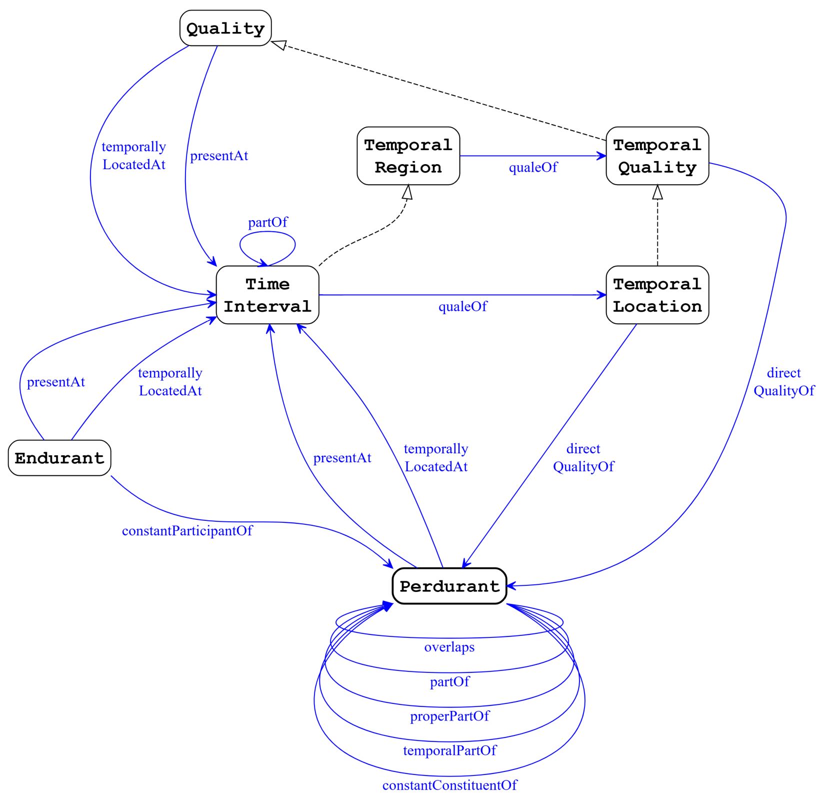

Figure 5: Object properties for `Perdurant` in DOLCEbasic$_{\mathrm{OWL}}$. Rounded boxes represent classes identified by their local IRI. Blue solid arrows denote object properties identified by their local IRIs, linking classes in their `rdfs:domain` to classes in their `rdfs:range`. Dashed arrows represent `rdfs:subClassOf` relations. Prefix `db:` is omitted for all classes and object properties.

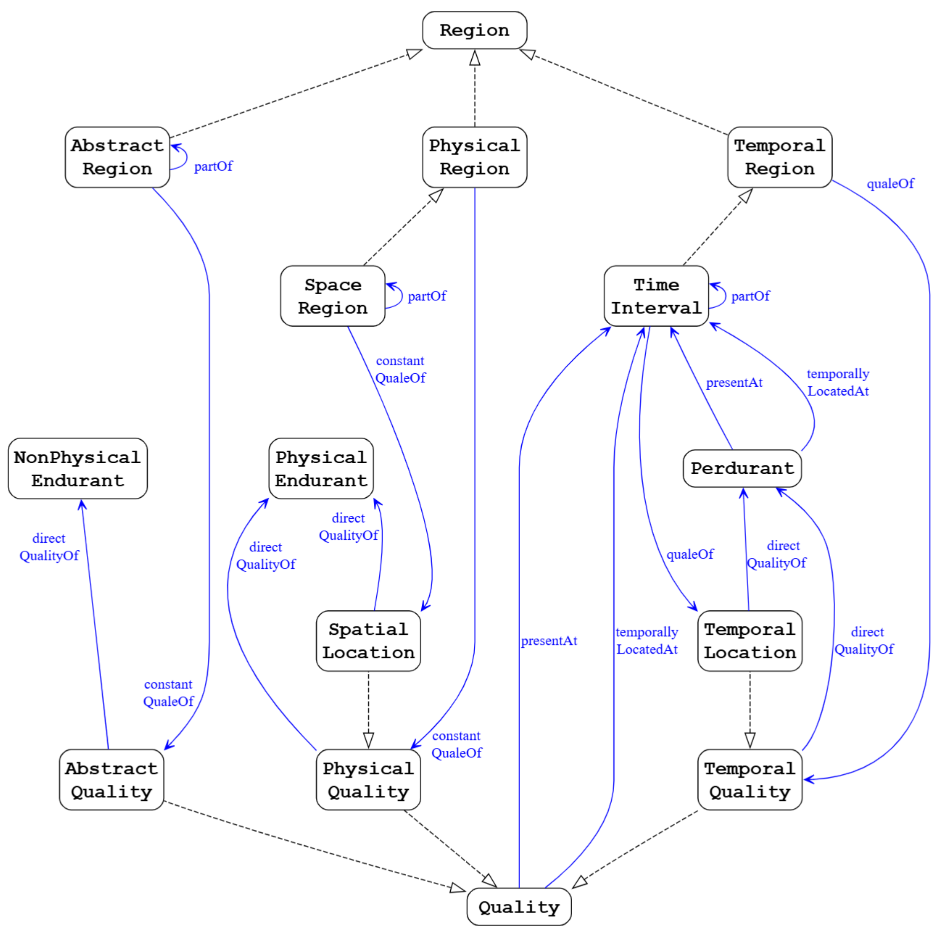

Figure 6: Object properties for `Quality` in DOLCEbasic$_{\mathrm{OWL}}$. Rounded boxes represent classes identified by their local IRI. Blue solid arrows denote object properties identified by their local IRIs, linking classes in their `rdfs:domain` to classes in their `rdfs:range`. Dashed arrows represent `rdfs:subClassOf` relations. Prefix `db:` is omitted for all classes and object properties.

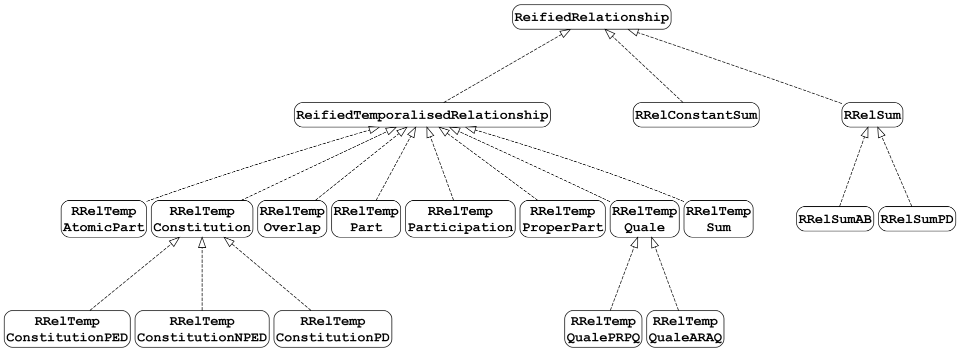

Figure 7: DOLCEnaryRel$_{\mathrm{OWL}}$ class hierarchy. Rounded boxes represent classes identified by their local IRI, while prefix `dn:` is omitted. Dashed arrows represent `rdfs:subClassOf` relations.

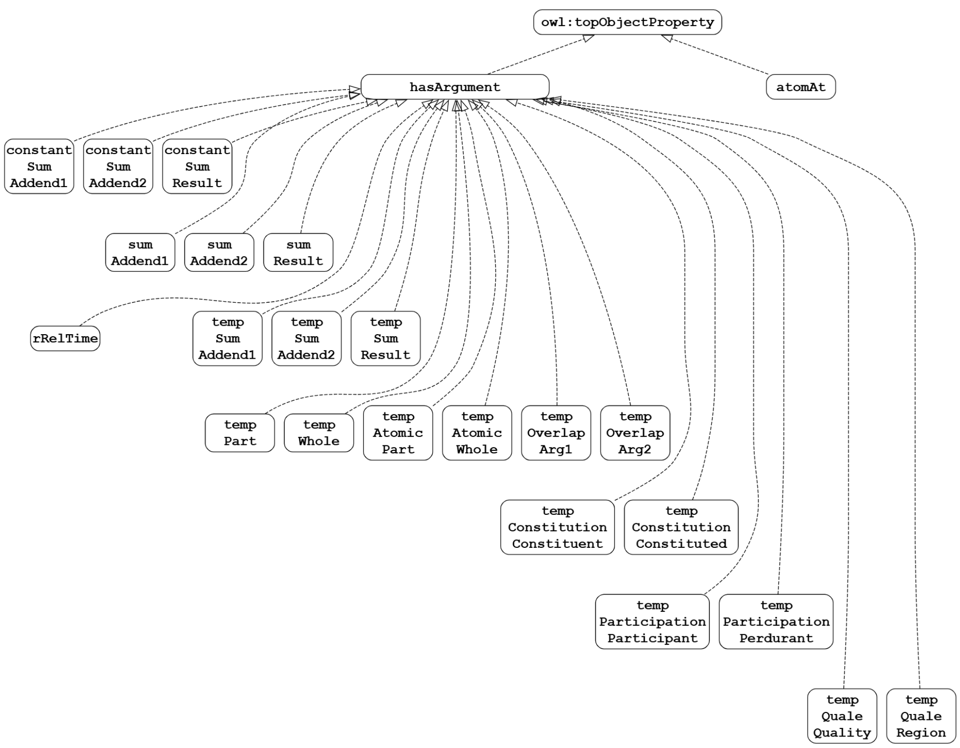

Figure 8: DOLCEnaryRel$_{\mathrm{OWL}}$ property hierarchy. Rounded boxes represent object properties identified by their local IRI, while prefix `dn:` is omitted. Dashed arrows represent `rdfs:subPropertyOf` relations.

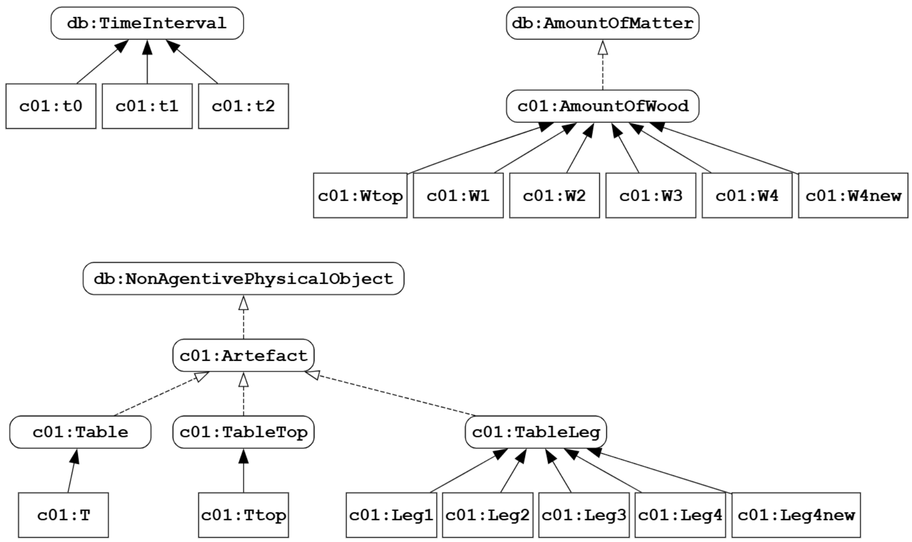

Figure 14: Graphical representation of individuals. Rounded boxes stand for classes, square boxes for instances, dashed arrows for `rdfs:subClassOf` relations, and solid arrows for `rdf:type` relations.

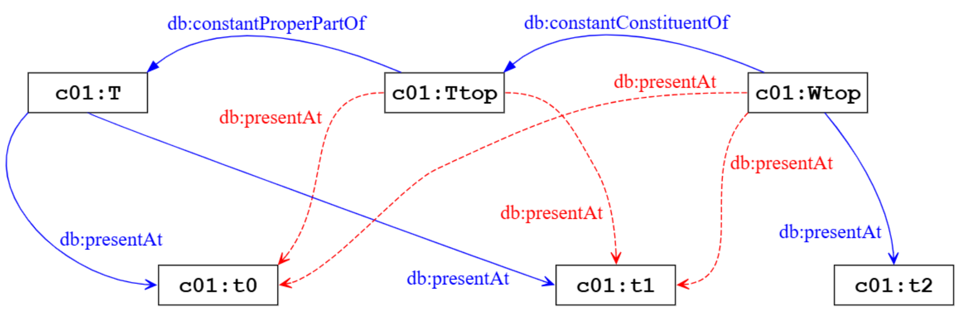

Figure 15: Graphical representation of constant relations. Square boxes stand for instances, blue solid arrows for relations labelled with the corresponding object property, and red dashed arrows for inferred relations.

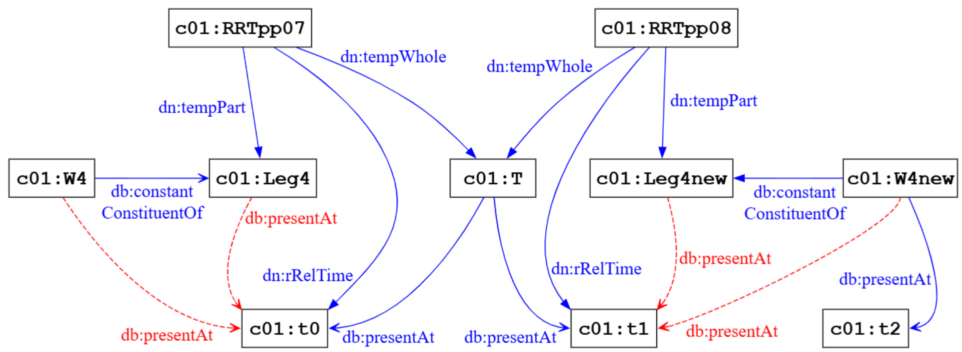

Figure 16: Graphical representation of temporalised relationships. Square boxes stands for instances, blue solid arrows for relations labelled with the corresponding object property, and red dashed arrows for inferred relations.

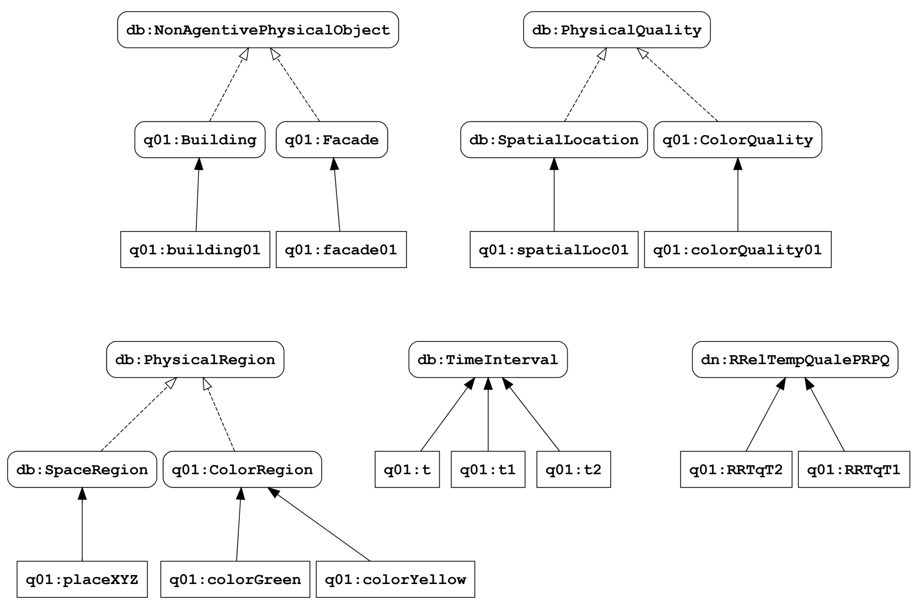

Figure 17: Graphical representation of individuals in the second use case. Rounded boxes stand for classes, square boxes for instances, dashed arrows for `rdfs:subClassOf` relations, and solid arrows for `rdf:type` relations.

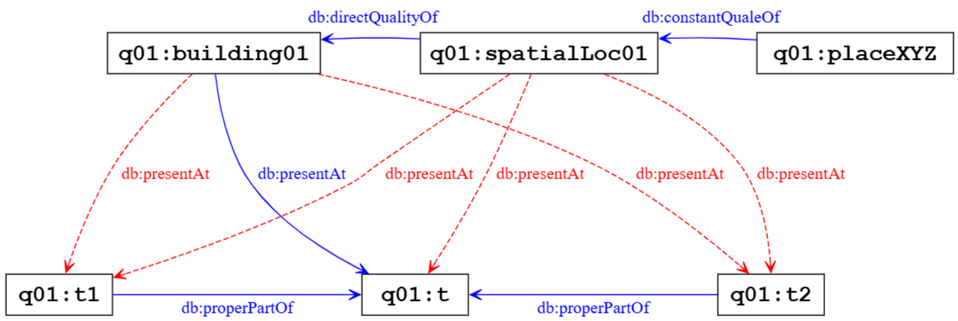

Figure 18: Graphical representation of constant relations. Square boxes stands for instances, blue solid arrows for relations labelled with the corresponding object property, and red dashed arrows for inferred relations.

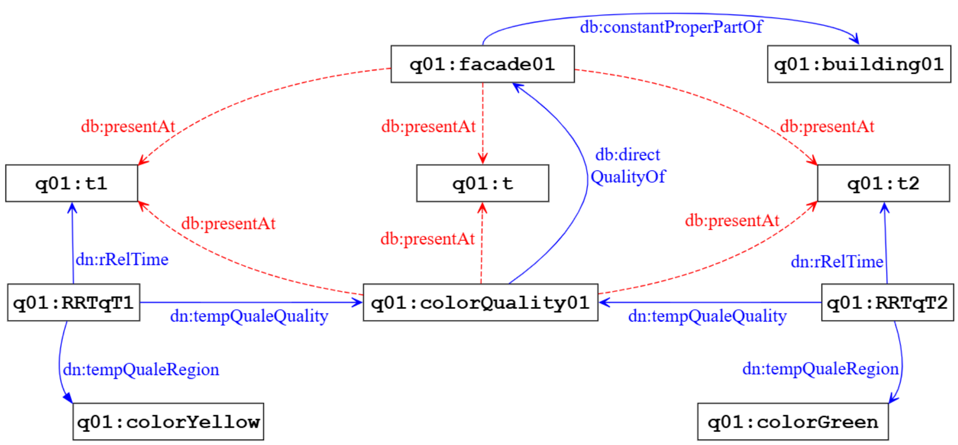

Figure 19: Graphical representation of temporalised relations. Square boxes stands for instances, blue solid arrows for relations labelled with the corresponding object property, and red dashed arrows for inferred relations.