

UFO: Unified Foundational Ontology

Giancarlo Guizzardi ^{a,b,*}, Alessander Botti Benevides ^c, Claudenir M. Fonseca ^a, Daniele Porello ^d, João Paulo A. Almeida ^c, and Tiago Prince Sales ^a

^a *Conceptual and Cognitive Modeling Research Group, Free University of Bozen-Bolzano, Italy*

E-mail: gguizzardi@unibz.it; cmoraisfonseca@unibz.it; tiago.princesales@unibz.it

^b *Services & Cybersecurity Group, University of Twente, The Netherlands*

^c *Ontology & Conceptual Modeling Research Group, Federal University of Espírito Santo, Brazil*

E-mail: abbenevides@inf.ufes.br; jpalmeida@ieee.org

^d *University of Genoa, Italy*

Email: daniele.porello@unige.it

Abstract: The Unified Foundational Ontology (UFO) was developed over the last two decades by consistently putting together theories from areas such as formal ontology in philosophy, cognitive science, linguistics, and philosophical logics. It comprises a number of micro-theories addressing fundamental conceptual modeling notions, including entity types and relationship types. The aim of this paper is to summarize the current state of UFO, presenting a formalization of the ontology, along with the analysis of a number of cases to illustrate the application of UFO and facilitate its comparison with other foundational ontologies in this special issue. (The cases originate from the First FOUST Workshop – the Foundational Stance, an international forum dedicated to Foundational Ontology research.)

Keywords: UFO, OntoUML, foundational ontology, ontological analysis, formal ontology

1. Introduction

For almost two decades now, the Unified Foundational Ontology (UFO) has been the focus of a long-term research program on ontological foundations for conceptual modeling. UFO (Guizzardi, 2005; Guizzardi et al., 2015b) was developed by consistently putting together theories from formal ontology in philosophy, cognitive science, linguistics, and philosophical logic. UFO is a four-category ontology that addresses fundamental conceptual modeling notions via a set of micro-theories, including:

- a theory of *types and taxonomic structures* that is connected to a theory of *object identifiers*, including a formal semantics in a sortal quantified modal logic (Guizzardi et al., 2021);
- a theory of *part-whole relations*;
- a theory of *particularized intrinsic properties, attributes and attribute value spaces* (Guizzardi and Zamborlini, 2014), which includes a view on datatypes as semantic reference structures (Albuquerque and Guizzardi, 2013);
- a theory of *particularized relational properties and relations*, (Guarino and Guizzardi, 2015; Fonseca et al., 2019), including a proposal for *Weak Truthmaking* (Guarino et al., 2019) connecting particularized properties to propositions;
- a theory of *roles*;
- a theory of *events* (Guizzardi et al., 2013; Almeida et al., 2019; Benevides et al., 2019; Guizzardi et al., 2016), including aspects such as event mereology, temporal ordering of events, object participation in events, causation, change, and the connection between events and endurants via dispositions;
- a theory for *multi-level modeling* (Guizzardi et al., 2015a; Fonseca et al., 2021a).

*Corresponding Author: E-mail: gguizzardi@unibz.it.

This program was initially inspired by seminal work on ontological foundations for conceptual modeling (Wand and Weber, 1989, 1990; Weber, 1997) that had employed the work of physicist and philosopher of science Mario Bunge (Bunge, 1977) to evaluate a number of conceptual modeling languages, including NIAM (Weber and Zhang, 1991), ER (Wand et al., 1999), UML (Evermann and Wand, 2001), and OWL (Bera and Wand, 2004). Despite the fruitfulness of the application of Bunge’s work to conceptual modeling, it soon became clear that there was a mismatch between the purposes for which Bunge’s ontology was developed and the requirements of ontological foundations for conceptual modeling. This became manifest in the literature as the predictions made by the Bunge Wand Weber (BWW) approach found themselves in strong contrast with the intuitions and practical knowledge of modelers, as well as with some predictions unanimously shared by alternative approaches (Guizzardi et al., 2015b). For example, BWW disavowed reified relationships, a position that conflicted with modeling predictions made by other foundational theories, such as Jackendoff’s semantic structures (Veres and Hitchman, 2002; Veres and Mansson, 2005), and with evidence from practitioners (Hitchman, 2003).

It was clear from the outset that an ontological theory for conceptual modeling would have to countenance both individuals and types, accounting for not only substantials and their types but also accidents and their types (e.g. particularized properties, moments, qualities, modes, tropes, abstract particulars, aspects, ways). In other words, a four-category ontology (Lowe and Lowe, 2006) was required. We needed particularized properties not only because they were of great importance in making sense of language and cognition (Parsons, 1990; Masolo et al., 2003; Davidson, 2001, Ch. 6), but because they would repeatedly appear in the discourse of conceptual modelers. Moreover, particularized relations (relationships) and particularized intrinsic properties (e.g., often represented by the so-called weak entities) are frequently modeled as bearers of other particularized properties in the practice of conceptual modeling. In fact, as demonstrated by Guizzardi and colleagues (see Guizzardi, 2005; Guizzardi et al., 2006; Guarino and Guizzardi, 2015), many conceptual modeling problems can hardly be solved without considering particularized properties.

In initial papers for this project, Guizzardi and Wagner attempted to employ the General Formal Ontology (GFO) and the General Ontology Language (GOL) being developed in Leipzig, Germany, as a reference theory (Heller and Herre, 2004; Herre et al., 2004). More or less at the same time, a strong cooperation was established between them and the Laboratory for Applied Ontology (LOA – Trento, Italy), which was concurrently developing the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE; Masolo et al., 2003). In this setting, their first attempt was to unify DOLCE and GFO to produce a reference foundational ontology for conceptual modeling, hence the name Unified Foundational Ontology (Guizzardi and Wagner, 2004). Both theories were philosophically sound, formally characterized, and based on the so-called Aristotelian Square, i.e., they were “four-category ontologies” (Lowe and Lowe, 2006).

UFO has since evolved, always focused on the requirements of the conceptual modeling discipline. In particular, ontological foundations for conceptual modeling would demand micro-theories to address its most fundamental constructs, namely, entity types and relationship types¹. So, any reference theory for conceptual modeling would need a rich theory of entity (object) types and a rich theory of domain (also called material) relations. In the case of the former, we needed something in the spirit of the ontology of universals underlying the OntoClean methodology (Guarino and Welty, 2009) in order to systematize a number of notions that were pervasive in the conceptual modeling literature (e.g., types, roles, phases or states, mixins) but for which there were no precise definitions or consensus (Guizzardi, 2005). In that respect, GFO’s theory of universals still does not recognize these notions and DOLCE does not include universal as a category (DOLCE was designed as an ontology of particulars). Regarding the latter, DOLCE still does not include a theory of particularized relational properties (relational qualities) and the GFO theory of relations is subject to the so-called Bradley’s Regress (Bradley, 1893) and, hence, it can only be instantiated by infinite (logical) models. This feature renders it ill-suited for most conceptual modeling

¹Hence, the name of the so-called Entity-Relationship approach that gives the name to the most important conference in conceptual modeling!

applications. Finally, there were many additional specific aspects needed for a general ontology for conceptual modeling that were not addressed by the existing approaches (e.g., attributes and datatypes, a rich theory of cognitively/linguistically motivated part-whole relations, higher-order types).

UFO's evolution has also been influenced by its applications over the years. It has been employed as a basis for analyzing, reengineering, and integrating many modeling languages and standards in different domains (e.g., UML, BPMN, ArchiMate), as well as for the development of core and domain ontologies in different areas (Guizzardi et al., 2015b). For instance, it has been successfully used to provide conceptual clarification in complex domains such as risk, services, trust, law, biodiversity, design science research, among many others (a more comprehensive list is provided in Section 5).

Of all UFO's applications, one deserves special attention, namely, the use of UFO in the design of an ontology-driven conceptual modeling language, which later came to be known as OntoUML (Guizzardi, 2005). This is a language that reflects the ontological micro-theories comprising UFO. The observation of the application of OntoUML over the years conducted by several groups in a variety of domains also amounted to a fruitful empirical source of knowledge regarding the language and its foundations². In this paper, we incorporate the latest developments of UFO arising out of these observations (Guizzardi et al., 2021). More specifically, the fine-grained distinctions among types (differentiating types according to meta-types such as *Kinds*, *Phases*, *Roles* and *Mixins*) that were before only applied to substantials are now applied to endurants in general (including thus *existentially dependent* endurants such as: *qualities*, e.g., the perceived value of the experience, the color of the apple; *modes*, e.g., Paul's Dengue Fever, Matteo's capacity of programming in Scratch; and *relators*, e.g., John and Mary's Marriage, Giovanni's Employment at the UN). These were needed to capture subtle aspects of domains such as value, service, economic exchange, and many others (Blums and Weigand, 2017; Carmo et al., 2017; Guarino and Guizzardi, 2015; Zamborlini et al., 2017), and are also employed in some cases presented in this paper.

This paper can be consumed as an introduction to UFO, as we cover its main tenets (Section 2), its formalization (Section 3), its application to relevant illustrative domains (Section 4), and its relevance in the field of applied ontology and conceptual modeling (Section 5). Alternatively, readers interested in comparing UFO's modeling approach with those of other foundational ontologies in this special issue may proceed directly to Section 4. As these examples are mainly represented by means of OntoUML models and all axioms in the paper are preceded by descriptions in natural language, the axioms can be largely skipped, with Section 3 serving as a reference of the axiomatization following the examples for those concerned with a more precise representation of UFO's approach.

2. Principles and structure of UFO

The objective of UFO was, from the beginning, to provide foundations for domain analysis in conceptual modeling, as well as for designing concrete models and modeling grammars. The idea was to develop a "Calculus of Content" aimed at supporting the ontological analysis, conceptual clarification, and semantic explication of the content embedded in representation artifacts. These, in turn, should support human users in tasks such as domain understanding, problem-solving, and meaning negotiation (Guarino et al., 2020).

As defended by the philosopher Kit Fine, "[...] *the fundamental question of ontology is not 'what is there?' but 'what is real?'* [...] *As a first stab, [...] what is real is what one must make reference to in giving a description of reality.*"³. In this spirit, UFO aims at providing instruments for describing what is real, at the mesoscopic level, and as accounted for by human cognition. For this reason, not only rocks and animals are real, but so are enrollments, employments, presidential mandates, symptoms, covalent bonds, birthday parties and football matches, colors and electrical charges, organizations and constitutions, software systems and derivative contracts.

²Several dozens of these models are available at <http://purl.org/krdb-core/model-repository/>.

³<https://www.3-16am.co.uk/articles/metaphysical-kit>

Given these objectives, UFO was constructed as a *descriptive* (as opposed to a *revisionary*) project (Strawson, 1959). This means that it is an ontology that takes into full consideration both cognitive and linguistic aspects informing its constituting categories. However, this is far from saying that reality is purely cognitively constructed. In this sense and, as put by Smith (2019), “[t]his is not to say that ontological items or entities—objects, properties, states of affairs, etc.—are ways of taking the world. Tables, detente, machinists, and love affairs are absolutely not merely epistemic entities... In that sense, [this view] is stubbornly realist: all these entities are things beyond us, things in the world. [They] are not ‘ways of taking the world’, They are the world taken a certain way.” From considering that “the world can be taken in many ways” (i.e., a pluralism of worldviews), it does not follow that anything goes, i.e., that all worldviews are equally valid or even acceptable. We should still “carve reality at its joints”, “joints” that are supplied by concrete causal nexus of properties associated with natural and nominal kinds (Keil, 1992). There are, however, multiple ontologically viable and cognitively useful ways of doing that. The ultimate goal of this project is, thus, providing well-founded engineering mechanisms for helping modelers to achieve *intra-worldview consistency*, i.e., ontological consistency when taking the world a certain way, and *inter-worldview interoperability*, i.e., making explicit the ontological commitments of a worldview such that different worldviews can safely interoperate.

UFO is organized in three main fragments:

- **UFO-A**, which is an ontology of endurants;
- **UFO-B**, which is an ontology of perdurants (Benevides et al., 2019);
- **UFO-C**, which is an ontology of social and intentional entities build on the foundations provided by the other two (de Oliveira Bringunte et al., 2011).

In this paper, we focus on the categories around UFO-A and a few categories of UFO-B, which are depicted in Figure 1. UFO is a 3D ontology having, as a fundamental distinction, the one between *endurants* (e.g., Mick Jagger, the Moon, John’s headache, Mary’s marriage to Paul) and *perdurants* (e.g., the 2020 U.S. presidential election, the World War II, the UEFA Euro 2021 Final), as opposed to a 4D ontology in which all concrete individuals are perdurants. Endurants are individuals that exist in time with all their parts. They have essential and accidental properties and, hence, they can qualitatively change while maintaining their numerical identity (i.e., while remaining the same individual). The sorts of changes an endurant can undergo while maintaining its identity is defined by the unique *Kind* it instantiates.

Perdurants⁴ are individuals that unfold in time accumulating temporal parts. They are manifestations of dispositions and only exist in the past. As such, perdurants are *modally fragile*, i.e., there is no cross-world identity between them and, hence, they cannot be in any way different than what they are (Guizzardi et al., 2016). Changes are perdurants but perdurants cannot be the subject of change. Any apparent change to a perdurant is either a variation (i.e., different temporal parts of an event having incompatible properties), or a change happening to some underlying endurant that is the focus of that event (Guizzardi et al., 2016). An endurant then *participates in* a perdurant if that perdurant has a part that is a *manifestation of a disposition inhering in* that endurant (Benevides et al., 2019).

Regarding the endurant fragment of the ontology, UFO is an ontology based of the so-called *Aristotelian Square*, thus, accounting for both substantial individuals (or *Substantials*), i.e., *independent entities* (e.g., Mick Jagger, the Free University of Bozen-Bolzano), as well as particularized properties, i.e., *existentially dependent entities* or *moments*, these last ones termed aspects, abstract particulars, or variable tropes (Moltmann, 2020). Moments are parasitic entities and can only exist by *inhering in* other entities.

Intrinsic moments are existentially dependent on a single individual. These include *qualities*, i.e., reifications of categorical properties such as color, height, weight, electrical charge. Qualities are entities that can be directly projected into certain value spaces. The latter, termed *quality structures*, are abstract entities delimiting the space of possible values (qualia, singular *quale*) for qualities of a given quality type. Quality structures can be either one-dimensional, in which case they are termed *quality dimensions*, or composed of multiple co-dependent dimensions (see also integral dimensions, Gärdenfors, 2004), in

⁴We often refer to perdurants as events, given that the latter is more easily understood by a non-philosophical audience.

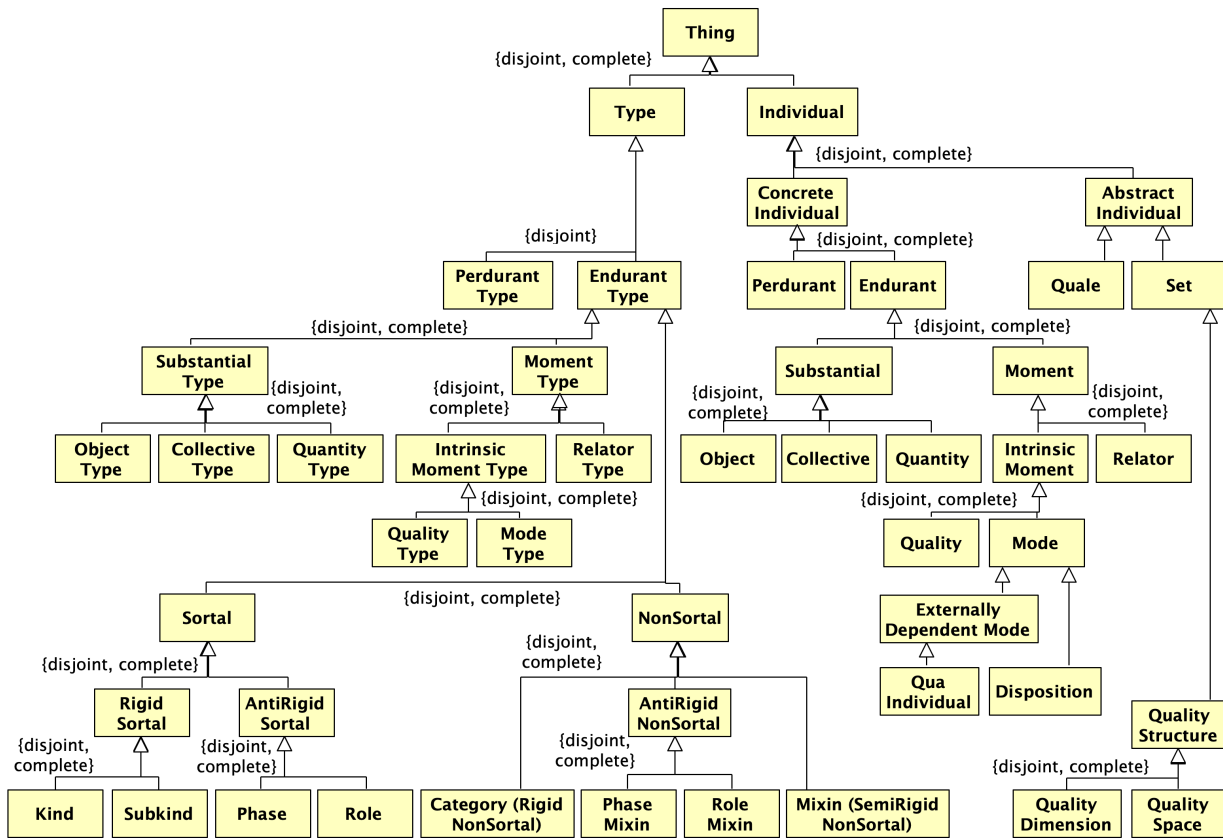


Figure 1. The Taxonomy of UFO.

which case they are termed *quality domains*. While the former includes height and weight dimensions as examples (structures isomorphic to a subset of the positive half-line of real numbers), the later includes as examples the color spindle and the taste tetrahedron.

Intrinsic moments also include *modes*. Modes can bear their own moments, including their own qualities, which can vary in independent ways. The category of modes include *dispositions* (e.g., functions, capabilities, capacities, vulnerabilities, etc.) as well as externally dependent entities (e.g., the love of John for Mary, the commitment of Paul towards Clara to meet for lunch next Friday). *Externally dependent modes* inhere in an entity while being *externally dependent* on another entity⁵. A particularly interesting type of externally dependent mode is a *qua individual*, which is a mode composed of other externally dependent modes that share the same bearer, the same source of external dependence, and the same *foundational event* (Guizzardi, 2005). For instance, the sum of all commitments and claims of John towards Mary form such a complex mode that we could call John-qua-husband-of-Mary, as they all inhere in John, are externally dependent of Mary, and are founded in that particular wedding event. Likewise, a *relator* is a moment (i.e., an existentially dependent entity) that is an aggregation of qua individuals (Guizzardi, 2005). For instance, John and Mary's Marriage is composed of John-qua-husband-of-Mary and Mary-qua-wife-of-John. A relator is, hence, existentially dependent on multiple individuals, namely, the bearers of its constituting qua individuals. Examples of relators include marriages, enrollments, employments, contracts, and presidential mandates. Externally dependent modes as well as relators serve as truthmarkers of material relations (Guarino and Guizzardi, 2015; Guarino et al., 2019), the first for one-sided relations and the second for double-sided (or multi-sided) ones (Fonseca et al., 2019).

Both substantials and moments are full-fledged endurants, being both able to bear moments, i.e., including cases involving moments of moments (e.g., the intensity of my headache, the hue of the color).

⁵That is, it is existentially dependent on an entity mereologically disjoint from the entity it inheres in (aka its bearer).

All endurants can have essential and accidental properties and can qualitatively change while maintaining their identity, provided they respect the principle of application provided by their respective Kinds. For example, John and Mary's marriage can change from a marriage with full separation of assets to one with partial separation of assets. The way their marriage can change is defined by the Kind it instantiates (marriage). Additionally, all endurants can have parts (e.g., George's employment to the UN is composed of a number of mutual claims and obligations).

Mereologically complex substantials are characterized by the different unity conditions that bind their parts together:

- *Quantities* are maximally-topologically-self-connected (e.g., pieces) of homeomerous amounts of matter, e.g., that puddle of water, this particular pile of sand;
- *Collectives* are entities whose parts play the same role with respect to the whole, e.g., the Black Forest (as a collective of trees), a deck of cards, the Dutch-speaking group in the crowd; and
- *Objects* (aka *functional complexes*) are entities whose parts play differentiated functional roles with respect to the whole, e.g., a human body, an organization, a computer, John's car.

UFO is an ontology of both particulars and universals (or types), i.e., patterns of features that are repeatable across different particulars. In fact, UFO is an ontology that also countenances high-order types, i.e., types whose instances are also types. The hierarchy of types in UFO reflects on the one hand the existing distinctions among sorts of individuals. So, we have perdurant types, substantial types, moment types, etc. On the other hand, UFO employs a number of formal meta-properties to spawn a rich typology of endurant types (Guizzardi et al., 2021).

As previously mentioned, a fundamental sort of endurant type is a *kind*, which provides uniform principles of individuation, identity, and persistence to its instances. For example, the types person, dog, computer, car, headache, organization and marriage are typically considered to be kinds. Kinds apply to instantiating individuals in all possible situations in which these individuals exist, and hence kinds are *rigid* types, applying necessarily to their instances. A *sortal* is either a kind or a specialization of a kind, and every sortal that is not a kind specializes exactly one kind. These specializations can be either themselves rigid, in which case they are termed *subkinds* (e.g., hatchback car as a subkind of car; financial organization as a subkind of organization) or they can be *anti-rigid*, i.e., classifying only contingently their instances. Among the latter, we have sortals whose contingent classification conditions are intrinsic, termed *Phases* (e.g., teenager as a phase of person, hemorrhagic dengue fever as a phase of dengue fever, and tenured employment as a phase of employment), and we have sortals whose contingent classification conditions are relational, termed *roles* (e.g., employee as a role of a person in the scope of an employment relator, and husband as a role of a person in the scope of a marriage relator).

In contrast with sortals, non-sortals are types that represent common properties of individuals of multiple Kinds. These can be:

- *categories*: rigid types that define essential properties for their instances, e.g., the category 'physical object' describing the properties of having a mass and a spatial extension, common to things of the kinds car, person, bridge, cow, etc.;
- *phase mixins*: anti-rigid types that define contingent properties for their instances. Their instantiation is characterized by *intrinsic contingent conditions*. For example, the phase mixin 'living animal' may apply to instances of the kinds person, dog, and horse; the phase mixin 'functional device' may characterize instances of the kinds computer, watch, and espresso machine;
- *role mixins*: anti-rigid types that define contingent properties for their instances. Their instantiation is characterized by *relational contingent conditions*. Examples include 'customer' for the kinds person and organization, but also 'insured legal relator' for the kinds employment and enrollment;
- *mixins*: semi-rigid types that define properties that are essential to some of their instances but accidental to some other instances (e.g., being a 'music artist' is essential to bands but accidental to people).

2.1. Summary

For the sake of comparison with other foundational ontologies within this special issue, we summarize UFO's core positions in the list below:

- I. UFO distinguishes between endurants and perdurants. Endurants are individuals that exist in time with all their parts. Perdurants are individuals that unfold in time accumulating temporal parts. An endurant can change while maintaining its identity, a perdurant cannot.
- II. UFO accounts for the existence of both independent and dependent endurants. Independent endurants are called substantials, while those that existentially depend on other individuals are dubbed moments. More precisely, a moment is said to inhere in another endurant. Perdurants are always dependent on the substantials that participate in them.
- III. Perdurants are past entities in UFO and cannot genuinely change. Alleged change in a perdurant is thus either a reflection of a perdurant having temporal parts with different properties, or, a change in an endurant (e.g., a mode or relator) of which the event is a manifestation. Thus, instead of admitting those entities that Stout refers to as "processes" (Stout, 1997), we rather consider those underlying endurants as the subject of change, and track their manifestations in time.
- IV. UFO accounts for (particularized) intrinsic and relational properties, which can serve as truth-makers for different types of relations and attributes. Particularized intrinsic properties that can be projected into a value space are deemed qualities, whilst those that cannot are deemed modes. Particularized relational properties are called relators. The term quantity is reserved for substantials that are maximally-topologically-self-connected of homeomerous amounts of matter.
- V. In UFO, roles are anti-rigid types whose instantiation occur in virtue of relational conditions. Roles that can be instantiated by individuals of a single kinds are simply termed roles, while those that can be instantiated by individuals of different kinds (thus, adhering to different identity principles) are termed roleMixins. There is no explicit definition of function in UFO.

3. The formalization of UFO in First-Order Modal Logic

In this section, we present a first-order modal theory of endurant types, in which types and their instances are both in the domain of quantification (i.e., first-order citizens), being connected by the instantiation relation (symbolized as " \vdash "). For our purposes, the first-order modal logic QS5 plus the Barcan formula and its converse suffices (Fitting and Mendelsohn, 2012). That means that we assume a *fixed* domain of entities for every possible world, what is traditionally associated to a *possibilistic* view of the entities of the domain, i.e., the domain includes all the *possibilia*. The modal operators of *necessity* (\Box) and *possibility* (\Diamond) are used with their usual meaning.

We drop both the universal quantifier and the necessity operator in case their scope takes the full formula. Definitions, introduced by " $\stackrel{def}{=}$ ", "are to be viewed as extraneous conventions of notational abbreviation" (Quine, 1937, p. 72). Axioms, theorems and definitions are indexed as in "(an)", "(tn)" and "(dn)," respectively. In a few cases when it is necessary to refer to the original formalization of UFO, we use "(m.n)" to denote the *n*th axiom of the *m*th chapter of Guizzardi (2005). Finally, this section is based on a first-order logic formalization of UFO⁶ specified in the TPTP syntax (Sutcliffe, 2017), and submitted to multiple automated provers for consistency and satisfiability checks.

⁶The complete first-order logic formalization of UFO in TPTP is available at <https://purl.org/ufo-formalization>.

3.1. Types, individuals and instantiation

Firstly, *types* are implicitly defined as those entities that are possibly instantiated (a1), while *individuals* are those necessarily not instantiated (a2). The domain of $::$ is $\text{Type} \cup \text{Individual}$ (a3) and it follows from (a1) that the codomain is Type . We deal here only with first-order and second-order types (a4). Axioms (a1) and (a2) entail that entities are partitioned into individuals and types (t1), (t2).

$$\begin{array}{ll}
 \mathbf{a1} & \text{Type}(x) \leftrightarrow \diamond(\exists y(y :: x)) \\
 \mathbf{a2} & \text{Individual}(x) \leftrightarrow \square(\neg\exists y(y :: x)) \\
 \mathbf{a3} & x :: y \rightarrow (\text{Type}(x) \vee \text{Individual}(x)) \\
 \mathbf{a4} & \neg\exists x, y, z(\text{Type}(x) \wedge x :: y \wedge y :: z) \\
 \mathbf{t1} & \text{Individual}(x) \vee \text{Type}(x) \\
 \mathbf{t2} & \neg\exists x(\text{Individual}(x) \wedge \text{Type}(x))
 \end{array}$$

We introduce the *specialization* relation between types (\sqsubseteq) defining it in terms of necessary extensional inclusion (a5), i.e., inclusion of their instances. Proper specialization (\sqsubset) is defined in the usual way (d1). By means of (a5), it follows that the specialization relation is quasi-reflexive (t3) and transitive (t4). Whenever two types have a common instance, they must share a supertype or a subtype for this instance (a6).

$$\begin{array}{ll}
 \mathbf{a5} & x \sqsubseteq y \leftrightarrow \text{Type}(x) \wedge \text{Type}(y) \wedge \square(\forall z(z :: x \rightarrow z :: y)) \\
 \mathbf{d1} & x \sqsubset y \stackrel{\text{def}}{=} x \sqsubseteq y \wedge \neg(y \sqsubseteq x) \\
 \mathbf{a6} & \forall t_1, t_2, x((x :: t_1 \wedge x :: t_2 \wedge \neg(t_1 \sqsubseteq t_2) \wedge \neg(t_2 \sqsubseteq t_1)) \rightarrow (\exists t_3(t_1 \sqsubseteq t_3 \wedge t_2 \sqsubseteq t_3 \wedge x :: t_3) \vee \exists t_3(t_3 \sqsubseteq t_1 \wedge t_3 \sqsubseteq t_2 \wedge x :: t_3))) \\
 \mathbf{t3} & x \sqsubseteq y \rightarrow (x \sqsubseteq x \wedge y \sqsubseteq y) \\
 \mathbf{t4} & x \sqsubseteq y \wedge y \sqsubseteq z \rightarrow x \sqsubseteq z
 \end{array}$$

Among the individuals, there are concrete individuals (a7) and abstract individuals (a8)–(a10). Concrete individuals are further differentiated into endurants (a11) and perdurants (a12). The classification of individuals into concrete and abstract individuals (a9), as well as the classification of concrete individuals into endurants and perdurants (a13)–(a14), are disjoint and complete.

$$\begin{array}{ll}
 \mathbf{a7} & \text{ConcreteIndividual}(x) \rightarrow \text{Individual}(x) \\
 \mathbf{a8} & \text{AbstractIndividual}(x) \rightarrow \text{Individual}(x) \\
 \mathbf{a9} & \text{ConcreteIndividual}(x) \rightarrow \neg\text{AbstractIndividual}(x) \\
 \mathbf{a10} & \text{Individual}(x) \leftrightarrow \text{ConcreteIndividual}(x) \vee \text{AbstractIndividual}(x) \\
 \mathbf{a11} & \text{Endurant}(x) \rightarrow \text{ConcreteIndividual}(x) \\
 \mathbf{a12} & \text{Perdurant}(x) \rightarrow \text{ConcreteIndividual}(x) \\
 \mathbf{a13} & \text{Endurant}(x) \rightarrow \neg\text{Perdurant}(x) \\
 \mathbf{a14} & \text{ConcreteIndividual}(x) \leftrightarrow \text{Endurant}(x) \vee \text{Perdurant}(x)
 \end{array}$$

Moreover, types are classified into types of endurants and types of perdurants (a15)–(a17), which are both disjoint (a17).

$$\begin{array}{l}
 \mathbf{a15} \quad \text{EndurantType}(x) \rightarrow \text{Type}(x) \\
 \mathbf{a16} \quad \text{PerdurantType}(x) \rightarrow \text{Type}(x) \\
 \mathbf{a17} \quad \text{EndurantType}(x) \rightarrow \neg\text{PerdurantType}(x)
 \end{array}$$

3.2. Some relevant type qualifications

UFO includes categories for types of endurants based on their rigidity and sortality. We implicitly define *rigidity* of endurant types as rigid (a18), anti-rigid (a19) and semi-rigid (a20), concluding that every endurant type is either one of the three (t5)–(t6). Also, rigid and semi-rigid types cannot specialize anti-rigid ones (t7)–(t8).

$$\begin{array}{l}
 \mathbf{a18} \quad \text{Rigid}(t) \leftrightarrow \text{EndurantType}(t) \wedge \forall x(\diamond(x :: t) \rightarrow \square(x :: t)) \\
 \mathbf{a19} \quad \text{AntiRigid}(t) \leftrightarrow \text{EndurantType}(t) \wedge \forall x(\diamond(x :: t) \rightarrow \diamond(\neg x :: t)) \\
 \mathbf{a20} \quad \text{SemiRigid}(t) \leftrightarrow \text{EndurantType}(t) \wedge \neg\text{Rigid}(t) \wedge \neg\text{AntiRigid}(t)
 \end{array}$$

- t5** $\text{EndurantType}(t) \leftrightarrow \text{Rigid}(t) \vee \text{AntiRigid}(t) \vee \text{SemiRigid}(t)$
t6 $\neg \exists x((\text{Rigid}(x) \wedge \text{AntiRigid}(x)) \vee (\text{Rigid}(x) \wedge \text{SemiRigid}(x)) \vee (\text{SemiRigid}(x) \wedge \text{AntiRigid}(x)))$
t7 $\neg \exists x, y(\text{Rigid}(x) \wedge \text{AntiRigid}(y) \wedge x \sqsubseteq y)$
t8 $\neg \exists x, y(\text{SemiRigid}(x) \wedge \text{AntiRigid}(y) \wedge x \sqsubseteq y)$

On *sortality*, our basic assumption is that every endurant necessarily instantiates a kind (a21), and everything necessarily instantiates at most one kind (a22). We implicitly define *sortals* as those endurant types whose instances necessarily instantiate the same kind (a23); while a *non-sortal* is an endurant type that is necessarily not a sortal (a24). As theorems, we have that: kinds are rigid (t9); kinds are necessarily disjoint (t10); a kind cannot specialize a different kind (t11); kinds are sortals (t12); sortals specialize a kind (t13); sortals cannot specialize different kinds (t14); a non-sortal cannot specialize a sortal (t15); and non-sortals do not have direct instances, their instances are also instances of some sortal that either specializes the non-sortal, or specializes a common non-sortal supertype (t16).

- a21** $\text{Endurant}(x) \rightarrow \exists k(\text{Kind}(k) \wedge \square(x :: k))$
a22 $\text{Kind}(k) \wedge x :: k \rightarrow \neg \diamond(\exists z(\text{Kind}(z) \wedge x :: z \wedge z \neq k))$
a23 $\text{Sortal}(t) \leftrightarrow \text{EndurantType}(t) \wedge \exists k(\text{Kind}(k) \wedge \square(\forall x(x :: t \rightarrow x :: k)))$
a24 $\text{NonSortal}(t) \leftrightarrow \text{EndurantType}(t) \wedge \neg \text{Sortal}(t)$
t9 $\text{Kind}(k) \rightarrow \text{Rigid}(k)$
t10 $\text{Kind}(x) \wedge \text{Kind}(y) \wedge x \neq y \rightarrow \square(\neg \exists z(z :: x \wedge z :: y))$
t11 $\text{Kind}(x) \wedge \text{Kind}(y) \wedge x \neq y \rightarrow (\neg(x \sqsubseteq y) \wedge \neg(y \sqsubseteq x))$
t12 $\text{Kind}(t) \rightarrow \text{Sortal}(t)$
t13 $\text{Sortal}(x) \rightarrow \exists k(\text{Kind}(k) \wedge x \sqsubseteq k)$
t14 $\neg \exists x, y, z(\text{Kind}(y) \wedge \text{Kind}(z) \wedge y \neq z \wedge x \sqsubseteq y \wedge x \sqsubseteq z)$
t15 $\neg \exists x, y(\text{NonSortal}(x) \wedge \text{Sortal}(y) \wedge x \sqsubseteq y)$
t16 $(\text{NonSortal}(t) \wedge x :: t) \rightarrow (\exists s(\text{Sortal}(s) \wedge s \sqsubseteq t \wedge x :: s) \vee \exists n, s(\text{NonSortal}(n) \wedge \text{Sortal}(s) \wedge s \sqsubseteq n \wedge t \sqsubseteq n \wedge x :: s))$

Regarding the leaves of the taxonomy of endurant types according to their sortality and rigidity, kinds and subkinds are disjoint (a25), and together encompass all rigid sortals (a26). Phases and roles are disjoint (a27), and together encompass all antirigid sortals (a28). Semi-rigid sortals are those that are semi-rigid and sortal (a29). Categories are those types that are rigid and non-sortal (a30). Mixins are those types that are semirigid and non-sortal (a31). Phase-mixins and role-mixins are disjoint (a32), and together encompass all antirigid non-sortals (a33). Let \mathcal{L}_T be the set of the leaf categories of the aforementioned taxonomy of endurant types {Kind, SubKind, Role, Phase, SemiRigidSortal, RoleMixin, PhaseMixin, Category, Mixin}, it follows that these leaf categories are pairwise disjoint (t17) and complete (t18).

- a25** $\neg \exists t(\text{Kind}(t) \wedge \text{SubKind}(t))$ **a29** $\text{SemiRigidSortal}(t) \leftrightarrow \text{SemiRigid}(t) \wedge \text{Sortal}(t)$
a26 $\text{Kind}(t) \vee \text{SubKind}(t) \leftrightarrow \text{Rigid}(t) \wedge \text{Sortal}(t)$ **a30** $\text{Category}(t) \leftrightarrow \text{Rigid}(t) \wedge \text{NonSortal}(t)$
a27 $\neg \exists t(\text{Phase}(t) \wedge \text{Role}(t))$ **a31** $\text{Mixin}(t) \leftrightarrow \text{SemiRigid}(t) \wedge \text{NonSortal}(t)$
a28 $\text{Phase}(t) \vee \text{Role}(t) \leftrightarrow \text{AntiRigid}(t) \wedge \text{Sortal}(t)$ **a32** $\neg \exists t(\text{PhaseMixin}(t) \wedge \text{RoleMixin}(t))$
a33 $\text{PhaseMixin}(t) \vee \text{RoleMixin}(t) \leftrightarrow \text{AntiRigid}(t) \wedge \text{NonSortal}(t)$
t17 $\bigwedge_{i, j \in \mathcal{L}_T, i \neq j} i(t) \rightarrow \neg j(t)$ **t18** $\text{EndurantType}(x) \leftrightarrow \bigvee_{i \in \mathcal{L}_T} i(x)$

3.3. Endurants

On the UFO taxonomy of endurants, Endurant is partitioned into Substantial and Moment (a34), (a35). Additionally, Substantial is partitioned into Object, Collective and Quantity (a36), (a37), (a38) and (a39).

Further, Moment is partitioned into Relator and IntrinsicMoment (a40), (a41). Finally, IntrinsicMoment is partitioned into Mode and Quality (a42), (a43). Let \mathcal{L}_e be the set of the leaf categories of endurants {Object, Collective, Quantity, Relator, Mode, Quality}, it is a theorem that these leaf categories partition Endurant (t19), (t20).

- a34** $\text{Substantial}(x) \vee \text{Moment}(x) \leftrightarrow \text{Endurant}(x)$
a35 $\neg \exists x(\text{Substantial}(x) \wedge \text{Moment}(x))$
a36 $\text{Object}(x) \vee \text{Collective}(x) \vee \text{Quantity}(x) \leftrightarrow \text{Substantial}(x)$
a37 $\neg \exists x(\text{Object}(x) \wedge \text{Collective}(x))$
a38 $\neg \exists x(\text{Object}(x) \wedge \text{Quantity}(x))$
a39 $\neg \exists x(\text{Collective}(x) \wedge \text{Quantity}(x))$
a40 $\text{Relator}(x) \vee \text{IntrinsicMoment}(x) \leftrightarrow \text{Moment}(x)$
a41 $\neg \exists x(\text{Relator}(x) \wedge \text{IntrinsicMoment}(x))$
a42 $\text{Mode}(x) \vee \text{Quality}(x) \leftrightarrow \text{IntrinsicMoment}(x)$
a43 $\neg \exists x(\text{Mode}(x) \wedge \text{Quality}(x))$
t19 $\bigwedge_{i,j \in \mathcal{L}_e, i \neq j} i(t) \rightarrow \neg j(t)$ **t20** $\text{Endurant}(x) \leftrightarrow \bigvee_{i \in \mathcal{L}_e} i(x)$

3.4. Endurant types

We define an orthogonal taxonomy of endurant types according to the ontological nature of their instances. Let \mathcal{P}_E be the set of pairs {(EndurantType, Endurant); (PerdurantType, Perdurant); (SubstantialType, Substantial); (MomentType, Moment)}; and $\mathcal{P}_{E'}$ be the set of pairs {(ObjectType, Object); (CollectiveType, Collective); (QuantityType, Quantity); (RelatorType, Relator); (ModeType, Mode); (QualityType, Quality)}. We implicitly define these types in the axiom schema (a44). It follows that the types in $\mathcal{P}_{E'}$ are pairwise disjoint (t21).

- a44** $\bigwedge_{(i,j) \in \mathcal{P}_E \cup \mathcal{P}_{E'}} i(t) \leftrightarrow \text{Type}(t) \wedge \Box(\forall x(x :: t \rightarrow j(x)))$
t21 $\bigwedge_{i,j \in \{k|(k, _) \in \mathcal{P}_{E'}\}, i \neq j} i(x) \rightarrow \neg j(x)$

Kinds are also specialized according to the ontological nature of their instances. Let \mathcal{P}_K be the set of pairs {(ObjectKind, ObjectType); (CollectiveKind, CollectiveType); (QuantityKind, QuantityType); (RelatorKind, RelatorType); (ModeKind, ModeType); (QualityKind, QualityType)}. We implicitly define these kinds in the axiom schema (a45). It is a theorem that all entities that possibly instantiate an endurant kind are endurants (t22). Moreover, every endurant instantiates one of the specific endurant kinds (a46). It follows that every endurant sortal is a type in $\mathcal{L}_{ES} = \{\text{ObjectKind, CollectiveKind, QuantityKind, RelatorKind, ModeKind, QualityKind, SubKind, Phase, Role, SemiRigidSortal}\}$ (t23); and that some sortals specialize specific kinds (t24).

- a45** $\bigwedge_{(i,j) \in \mathcal{P}_K} i(t) \leftrightarrow j(t) \wedge \text{Kind}(t)$
t22 $\diamond(\exists k((\text{ObjectKind}(k) \vee \text{CollectiveKind}(k) \vee \text{QuantityKind}(k) \vee \text{RelatorKind}(k) \vee \text{ModeKind}(k) \vee \text{QualityKind}(k)) \wedge x :: k)) \rightarrow \text{Endurant}(x)$
a46 $\text{Endurant}(x) \rightarrow \diamond(\exists k((\text{ObjectKind}(k) \vee \text{CollectiveKind}(k) \vee \text{QuantityKind}(k) \vee \text{RelatorKind}(k) \vee \text{ModeKind}(k) \vee \text{QualityKind}(k)) \wedge x :: k))$
t23 $\text{EndurantType}(x) \wedge \text{Sortal}(x) \rightarrow \bigvee_{i \in \mathcal{L}_{ES}} i(x)$
t24 $(\text{Sortal}(t) \wedge (\text{ObjectType}(t) \vee \text{CollectiveType}(t) \vee \text{QuantityType}(t) \vee \text{RelatorType}(t) \vee \text{ModeType}(t) \vee \text{QualityType}(t))) \leftrightarrow \exists k((\text{ObjectKind}(k) \vee \text{CollectiveKind}(k) \vee \text{QuantityKind}(k) \vee \text{RelatorKind}(k) \vee \text{ModeKind}(k) \vee \text{QualityKind}(k)) \wedge t \sqsubseteq k)$

We have as theorems that the leaves of the taxonomy of endurant types— $\mathcal{L}_{ET} = \{\text{ObjectKind, CollectiveKind, QuantityKind, SubKind, RelatorKind, ModeKind, QualityKind, SemiRigidSortal, Category, Phase, Mixin, Role, PhaseMixin, RoleMixin}\}$ —are pairwise disjoint (t25); and that every endurant type is a type in \mathcal{L}_{ET} (t26).

$$\text{t25} \quad \bigwedge_{i,j \in \mathcal{L}_{ET}, i \neq j} i(t) \rightarrow \neg j(t) \qquad \text{t26} \quad \text{EndurantType}(x) \rightarrow \bigvee_{i \in \mathcal{L}_{ET}} i(x)$$

3.5. Mereology

In the axioms (a47)–(a52) we formalize general extensional mereology (see Hovda, 2009). We present here the main axioms and definitions; for the purposes of the subsequent definitions, that suffices.

$$\begin{array}{ll} \text{a47} & P(x, x) \qquad \qquad \qquad \text{(reflexivity)} \\ \text{a48} & P(x, y) \wedge P(y, x) \rightarrow x = y \qquad \text{(anti-symmetry)} \\ \text{a49} & P(x, y) \wedge P(y, z) \rightarrow P(x, z) \qquad \text{(transitivity)} \\ \text{a50} & O(x, y) \leftrightarrow \exists z(P(z, x) \wedge P(z, y)) \qquad \text{(Overlap)} \\ \text{a51} & \neg P(y, x) \rightarrow \exists z(P(z, y) \wedge \neg O(z, x)) \qquad \text{(Strong supplementation)} \\ \text{a52} & PP(x, y) \leftrightarrow P(x, y) \wedge \neg P(y, x) \qquad \text{(Proper Part)} \end{array}$$

3.6. Composition

We summarize below the treatment of functional parts by Guizzardi (2005). We introduce a relation $F(x, x')$ between an object x and a type x' , whose meaning is that “ x functions as x' .” The relation of *generic functional dependence* between types is then defined as follows.

$$\text{a53} \quad \text{gfd}(x', y') \leftrightarrow \forall x(x :: x' \wedge F(x, x') \rightarrow \exists y(\neg(y = x) \wedge y :: y' \wedge F(y, y')))$$

We define the functional dependence between individuals x and y of types x' and y' (respectively) as follows.

$$\text{a54} \quad \text{ifd}(x, x', y, y') \leftrightarrow \text{gfd}(x', y') \wedge x :: x' \wedge y :: y' \wedge (F(x, x') \rightarrow F(y, y'))$$

We can now introduce the notion of *component of* that we are going to use in the subsequent example.

$$\text{a55} \quad \text{componentOf}(x, x', y, y') \leftrightarrow xPPy \wedge \text{ifd}(x, x', y, y')$$

3.7. Constitution

We adopt here a very simple and preliminary view of constitution. First, it is a relation that holds between things of the same ontological category (a56).

$$\text{a56} \quad \text{constitutedBy}(x, y) \rightarrow ((\text{Endurant}(x) \leftrightarrow \text{Endurant}(y)) \wedge (\text{Perdurant}(x) \leftrightarrow \text{Perdurant}(y)))$$

However, following Baker (2007), we have that constitution holds between things of different Kinds (a57). From this, we have the non-reflexivity of constitution as a theorem (t27). Notice that by instantiating different kinds, constituent and constituted obey different principles of identity, and have different modal properties (including different essential parts).⁷

$$\begin{array}{l} \text{a57} \quad \text{constitutedBy}(x, y) \wedge (x :: x') \wedge (y :: y') \wedge \text{Kind}(x') \wedge \text{Kind}(y') \rightarrow x' \neq y' \\ \text{t27} \quad \neg \text{constitutedBy}(x, x) \end{array}$$

⁷Thomson (1998) requires the following condition to hold in constitution: “there must be at least one essential part of the constituent, and no part of the constituent is essential to the constituted”.

Any complete theory of constitution should explain why material properties of the constituent are systematically related to those of the constituted. Typically, for physical endurants, the spatial location of the former are inherited by the latter⁸; for perdurants, the temporal properties of the former are constrained by the latter⁹. However, these derived relations can apply to many other properties, e.g., the weight, volume, color, chemical structure of the statue are derived from the properties of the constituting clay (Baker, 2007). Now, it would be hard to defend that these derivation relations work the other way around as well (e.g., that the weight, volume, color, chemical structure of the lump of clay are derived from the properties of the constituted statue). In fact, although it is natural to conceptualize *The Beatles* as being constituted by collective of John, Paul, George, Ringo, it is utterly unnatural to conceptualize the collective as being constituted by *The Beatles*. Why is that? We believe that this is because constitution relies on a notion of *grounding*. This seems to also explain requirements such as the one proposed by Baker (2007) that *special circumstances* must hold for the constituent to constitute the constituted (e.g., for the clay to constitute the statue, an intentional act of the sculptor is required), and that these circumstances are both necessary and sufficient.

This asymmetry is also often framed in terms of asymmetric cases of dependence (again, strongly connected to grounding). For example, every instance of statue of clay must be constituted by a particular lump of clay, but not the other way around, i.e., lumps of clay can exist without constituting statues. In other words, statues of clay are *generically dependent* on the type Lump of Clay. This can be captured with the notions of Generic Constitutional Dependence (GCD), between types x' and y' , as well as the type-level relation of Constitution (a59) between x of type x' , and y of type y' .

$$\mathbf{a58} \quad \text{GCD}(x', y') \leftrightarrow \forall x(x :: x') \rightarrow \exists y(y :: y') \wedge \text{constitutedBy}(x, y)$$

$$\mathbf{a59} \quad \text{Constitution}(x, x', y, y') \leftrightarrow (x :: x') \wedge (y :: y') \wedge \text{GCD}(x', y') \wedge \text{constitutedBy}(x, y)$$

Now, using another example, we can have a Boxing Match being constituted by one or more punches (again, an asymmetric dependence). In this case, however, a specific Boxing Match is constituted by specific punching events, i.e., a case of *specific dependence*. This is because we are dealing here with a constitution relation between events and, as previously discussed, events are modally fragile entities and, hence, all their parts and constituents are necessary constituents (a60).

$$\mathbf{a60} \quad \forall x, y \text{Perdurant}(x) \wedge \text{constitutedBy}(x, y) \rightarrow \Box(\text{ex}(x) \rightarrow \text{constitutedBy}(x, y))$$

In light of these notions of dependence, constitution seems to be similar to the relation of functional parthood (*componentOf*). *ComponentOf* is a material relation that also requires “special circumstances” in order to hold. Due to the different configurations involving different types of dependence relations holding between functional parts, functional parthood is not irrestrictively transitive. In fact, transitivity only holds in certain scopes defined by these “special circumstances”. Maybe an analogous case can be made to explain why transitivity does not seem to hold for constitution irrestrictively either. For example, the human tissue that constitutes Paul McCartney does not constitute *The Beatles*. Perhaps, like parthood (Guizzardi, 2005), constitution is not a single relation but a family of relations, each requiring additional axioms extending a very minimal common core.

Again, we advocate that a complete theory of constitution requires a proper theory of grounding. For this reason, we postpone proposing a complete theory for the former relation in UFO until we can fully advance a complete theory of the latter relation. In any case, to honor this strong intuition based on grounding (which is asymmetric) and asymmetric dependence, we assume here that constitution is an asymmetric relation (a61).

$$\mathbf{a61} \quad \text{constitutedBy}(x, y) \rightarrow \neg \text{constitutedBy}(y, x)$$

⁸We do admit, however, relations of constitution between non-physical endurants. For example, a software program is constituted by software code, not identical to it.

⁹For the case of parthood between events (Benevides et al., 2019), we have that the temporal interval framing the parts are part of the one framing the whole. *Mutatis mutandis*, for the case of constitution between events, we assume the same here for the relation between the temporal interval framing the constituent and the one framing the constituted.

In summary, in this rather simplified form, the general relation of constitution is taken here as a non-reflexive and asymmetric relation. Moreover, it is a relation that holds between entities of the same ontological categories but between entities of different Kinds. Specific types of constituents and constituted entities can extend this minimal axiomatization, for example, to constraint relations between properties of physical endurants or events. A fuller theory of derivation of properties in this sense, however, waits for a fuller theory of grounding.

3.8. Existence and Existential Dependence

UFO introduces an *existence predicate* defined on any possible entity. In principle, we can view existence as related to time, $\text{ex}(x, t)$, however, as we are proposing here a single time slice formalization of UFO, we introduce in (a62) existence as a unary predicate $\text{ex}(x)$. By means of the existence predicate, we implicitly define in (a63) the relation of *existential dependence* between two entities, $\text{ed}(x, y)$. We also implicitly define the notion of existential independence, $\text{ind}(x, y)$, accordingly (a64).

$$\mathbf{a62} \quad \text{ex}(x) \rightarrow \text{Thing}(x)$$

$$\mathbf{a63} \quad \text{ed}(x, y) \leftrightarrow \Box(\text{ex}(x) \rightarrow \text{ex}(y))$$

$$\mathbf{a64} \quad \text{ind}(x, y) \leftrightarrow \neg \text{ed}(x, y) \wedge \neg \text{ed}(y, x)$$

3.9. Moments and inherence

We summarize a few points of the formalization of substantials and moments of the original UFO, as they shall be required in the subsequent sections. Moments are known as *variable tropes*, abstract particulars, or particular qualities in the philosophical literature (Guizzardi, 2005). In UFO, moments can be viewed either as individualized properties, such as the color or the weight of an object, for the case of intrinsic moments, or a marriage or an enrollment, for the case of relational moments (relators), cf. (a40), (a41). The relation that connects moments to the object that they are about is the relation of *inherence*. Inherence is a type of existential dependence relation (a65) holding between a moment and an entity of which it depends, called its *bearer* (a66).

$$\mathbf{a65} \quad \text{inheresIn}(x, y) \rightarrow \text{ed}(x, y) \tag{6.4}$$

$$\mathbf{a66} \quad \text{inheresIn}(x, y) \rightarrow \text{Moment}(x) \wedge (\text{Type}(y) \vee \text{ConcreteIndividual}(y)) \tag{6.10}$$

Moreover, a moment cannot inhere in two separate individuals (a67), this is the so-called *non-migration* or *non-transferability* principle (a67).

$$\mathbf{a67} \quad \text{inheresIn}(x, y) \wedge \text{inheresIn}(x, z) \rightarrow y = z \tag{6.9}$$

Moments can also be involved in chains of inherence relations (d2), which are ultimately grounded on a unique entity that does not inhere in anything else. This entity that we will call here the *Ultimate Bearer*. (d3). Furthermore, the ultimate bearer of a moment is unique (a68).

$$\mathbf{d2} \quad \text{momentOf}(m, x) \stackrel{\text{def}}{=} \text{inheresIn}(m, x) \vee \exists y(\text{inheresIn}(m, y) \wedge \text{momentOf}(y, x))$$

$$\mathbf{d3} \quad \text{ultimateBearerOf}(b, m) \stackrel{\text{def}}{=} \neg \text{Moment}(b) \wedge \text{momentOf}(m, b)$$

$$\mathbf{a68} \quad \text{Moment}(m) \rightarrow \exists! b \text{ ultimateBearerOf}(b, m)$$

Finally, in line with what was proposed originally in UFO (Guizzardi, 2005), we have the following theorems (t28)–(t30), i.e., inherence is a non-reflexive, asymmetric and anti-transitive relation.

$$\mathbf{t28} \quad \neg \text{inheresIn}(x, x) \tag{6.5}$$

$$\mathbf{t29} \quad \text{inheresIn}(x, y) \rightarrow \neg \text{inheresIn}(y, x) \tag{6.6}$$

$$\mathbf{t30} \quad \text{inheresIn}(x, y) \wedge \text{inheresIn}(y, z) \rightarrow \neg \text{inheresIn}(x, z) \tag{6.7}$$

3.10. Relators

We summarize below the presentation of relators by Guizzardi (2005). Recent investigations on relators, relations, and their connections to perdurants (events) have been proposed by Guarino and Guizzardi (2015, 2016). We start by defining *externally dependent modes* (a70): a mode x that is existentially dependent on an entity that is independent of the bearer of x .

$$\begin{aligned} \text{a69} \quad & \text{externallyDependent}(x, y) \leftrightarrow \text{ed}(x, y) \wedge \forall z (\text{inheresIn}(x, z) \rightarrow \text{ind}(y, z)) \\ \text{a70} \quad & \text{ExternallyDependentMode}(x) \leftrightarrow \text{Mode}(x) \wedge \exists y (\text{externallyDependent}(x, y)) \end{aligned} \quad (6.42)$$

Externally dependent modes indeed capture truly relational qualities. For instance, by being married to Mary, in virtue of this relation, John acquires a number of properties (i.e. *modes*) that depend on John's relationship with Mary; John's commitment towards Mary is a mode that depends on an individual that is existentially independent of the bearer of John's commitment. The condition of existential independence excludes that an externally dependent mode may depend on individuals that are ontologically too close to the bearer of the mode. For instance, the courage of John is a mode that depends on John, who is not existentially independent of himself; for this reason, the courage of John is a mode that is not externally dependent.

Externally dependent modes (as well as relators, as we discuss later) are founded by means of a unique event (a71), (a72). E.g., John's conjugal commitments towards Mary are founded on the event of the wedding between John and Mary. Since every externally dependent mode is founded by an unique event, we can define $\text{foundationOf}(x)$ by means of a definite description (d4).

$$\begin{aligned} \text{a71} \quad & \text{foundedBy}(x, y) \rightarrow (\text{ExternallyDependentMode}(x) \vee \text{Relator}(x)) \wedge \text{Perdurant}(y) \\ \text{a72} \quad & \text{ExternallyDependentMode}(x) \rightarrow \exists! y (\text{foundedBy}(x, y)) \\ \text{d4} \quad & \text{foundationOf}(x) = y \leftrightarrow \text{foundedBy}(x, y) \end{aligned}$$

We can now explain what it means for an individual x to be a *qua individual* of another individual y : the relation qualIndividualOf holds between x and y iff x is the sum of all externally dependent modes of y that share the same foundational event (a73). Moreover, an entity is a *qua individual* iff it is a qualIndividualOf another entity (a74). For instance, in virtue of the marriage with Mary, John is the bearer of a number of externally dependent modes related to conjugal commitments that constitute John-qua-husband-of-Mary. For this reason, we view a qua individual as a complex externally dependent mode (a75), and so it has a unique foundation, which is the same as the foundation of its parts (a31). Finally, qua individuals are genuine individuals in the sense that they cannot be attached to two distinct particulars (a76). E.g., John qua student is the qua individual of John, not of any other person.

$$\begin{aligned} \text{a73} \quad & \text{qualIndividualOf}(x, y) \leftrightarrow \forall z (\text{O}(z, x) \leftrightarrow (\text{ExternallyDependentMode}(z) \wedge \text{inheresIn}(z, y) \wedge \\ & \text{foundationOf}(z) = \text{foundationOf}(x))) \\ \text{t31} \quad & \text{qualIndividualOf}(x, y) \wedge x' P x \rightarrow \text{foundationOf}(x) = \text{foundationOf}(x') \\ \text{a74} \quad & \text{QualIndividual}(x) \leftrightarrow \exists y \text{qualIndividualOf}(x, y) \\ \text{a75} \quad & \text{QualIndividual}(x) \rightarrow \text{ExternallyDependentMode}(x) \\ \text{a76} \quad & \text{qualIndividualOf}(x, y) \wedge \text{qualIndividualOf}(x, y') \rightarrow y = y' \end{aligned}$$

We assume that every relator is founded on a unique event (a77), which is also the foundation of the parts of the relator. Relators are then implicitly defined by (a79): they are sums of qua individuals that share the same foundation and are existentially dependent on each other.

$$\begin{aligned} \text{a77} \quad & \text{Relator}(x) \rightarrow \exists! y (\text{foundedBy}(x, y)) \\ \text{a78} \quad & \text{Relator}(x) \wedge y P x \rightarrow \text{foundationOf}(x) = \text{foundationOf}(y) \end{aligned}$$

$$\begin{aligned} \mathbf{a79} \quad \text{Relator}(x) \leftrightarrow \exists y(\text{PP}(y, x)) \wedge \forall y, z((\text{PP}(y, x) \wedge \text{PP}(z, x)) \rightarrow (\text{QualIndividual}(y) \wedge \\ \text{QualIndividual}(z) \wedge (\text{foundationOf}(y) = \text{foundationOf}(z)) \wedge \text{ed}(y, z) \wedge \text{ed}(z, y))) \wedge \\ \forall y, z((\text{PP}(y, x) \wedge \text{QualIndividual}(z) \wedge (\text{foundationOf}(y) = \text{foundationOf}(z)) \wedge \text{ed}(y, z) \wedge \\ \text{ed}(z, y)) \rightarrow \text{PP}(z, x)) \end{aligned}$$

For example, John and Mary's wedding is the foundation of a number of externally dependent modes of John that depend on Mary and of Mary that depend on John, the sum of which constitutes the marriage relator.

According to this view, relators bring about qua individuals. By (a79), there must exist at least two qua individuals, x' and x'' , that are part of the relator, given by the existence of a proper part in (a79) and by strong supplementation (a51). By (t32), those qua individuals are qua individuals of some other individual, y' and y'' . Thus, we establish (t32).

$$\mathbf{t32} \quad \text{Relator}(x) \rightarrow \exists x', x'', y', y''(\text{qualIndividualOf}(x', y') \wedge \text{qualIndividualOf}(x'', y''))$$

We introduce the relation of *mediation*, $\text{mediates}(x, y)$, between a relator x and an individual y that the relator connects in a relational statement (a80).

$$\mathbf{a80} \quad \text{mediates}(x, y) \leftrightarrow \text{Relator}(x) \wedge \text{Endurant}(y) \wedge \exists z.(\text{qualIndividualOf}(z, y) \wedge \text{P}zx)$$

$$\mathbf{t33} \quad \text{Relator}(x) \rightarrow \exists y, z(y \neq z \wedge \text{mediates}(x, y) \wedge \text{mediates}(x, z))$$

By axiom (a79), any relator has at least two distinct qua individuals as parts. Those qua individuals are associated to distinct individuals by (a74) and (a76), which are mediated by the relator by (a80). Thus, we infer (t33).¹⁰

Since a relator is a particular type of moment, it has to have a unique bearer, we define the bearer of the relator as the mereological sum of the individuals that the relator mediates. Moreover, a relator type is a type that applies to relators. For instance, *marriage* is a relator type of which the marriage between John and Mary is an instance. Relator types are defined according to schema (a44).

3.11. Characterization

Guizzardi (2005) states that a type is *characterized* by moment types that inhere in its instances.

$$\begin{aligned} \mathbf{a81} \quad \text{characterization}(t, m) \rightarrow \text{EndurantType}(t) \wedge \text{MomentType}(m) \wedge \\ \forall x.(x :: t \rightarrow \exists y.(y :: m \wedge \text{inheresIn}(y, x))) \wedge \forall z.(z :: m \rightarrow \exists! w.(w :: t \wedge \text{inheresIn}(z, w))) \end{aligned}$$

In particular, for quality kinds, we have that:

$$\mathbf{a82} \quad \text{characterization}(t, q) \wedge \text{QualityType}(q) \rightarrow (\forall x :: q \rightarrow \exists! y :: t \wedge \text{inheresIn}(x, y))$$

3.12. Qualities and quality structures

In the following axioms, we use the standard notation from set-theory and arithmetic, i.e., membership \in , set inclusion \subseteq , proper set inclusion \subset , Cartesian product \times , the empty set \emptyset , addition $+$, the greater or equal relation \geq , and the natural number zero 0 , by appealing to their intuitive meaning. However, we will not commit to a specific theory.¹¹

UFO grounds *quality structures* in its taxonomy of abstract individuals, which is classified into sets and quales (a83)–(a85).

¹⁰Our theorem (t33) corresponds to axiom (6.45) in Guizzardi's book (Guizzardi, 2005, §6.45). It is now a theorem, since we introduced a stronger definition of relators.

¹¹A suitable set theory should be enough for expressing these notions. For instance, New Foundations with Urelements (NFU) + Infinity + Choice can represent first-order Peano arithmetic (PA) and Cartesian product. One problem here is that theories of arithmetic, like Robinson arithmetic (Q) and PA, have no finite models.

- a83** $\text{Quale}(x) \rightarrow \text{AbstractIndividual}(x)$ **a85** $\neg \exists x(\text{Quale}(x) \wedge \text{Set}(x))$
a84 $\text{Set}(x) \rightarrow \text{AbstractIndividual}(x)$

A *quality structure* is defined as an entity associated with a unique *quality type* (d5). Moreover, quality structures are non-empty sets (a86). The members of quality structures are *quales*, where an entity is a quale iff it is a member of a unique quality structure (a87). A quality structure is partitioned into *quality dimension* and *quality domain* (a88), (a89). Finally, given two quality types such that one properly specializes the other, the sub-quality-type imposes constraints on the quality structure associated with the super-quality-type. Since quality structures are sets, the quality structure associated with the sub-quality-type must be a proper subset of the quality structure associated with the super-quality-type. For instance, given that Color is associated with ColorSpace, then SkinColor (a subkind of Color) is associated with a proper subset of ColorSpace. More formally, for any quality structures s and s' and quality types t and t' such that s is associated with t , s' is associated with t' , and t' is a proper specialization of t , then s' is a proper subset of s (a90).

- d5** $\text{QualityStructure}(x) \stackrel{def}{=} \exists!t(\text{QualityType}(t) \wedge \text{associatedWith}(x, t))$ (6.26)
a86 $\text{QualityStructure}(x) \rightarrow \text{Set}(x) \wedge x \neq \emptyset$ (6.27)
a87 $\text{Quale}(x) \leftrightarrow \exists!y(\text{QualityStructure}(y) \wedge x \in y)$ (6.40)
a88 $\text{QualityStructure}(x) \leftrightarrow \text{QualityDomain}(x) \vee \text{QualityDimension}(x)$ (6.23)
a89 $\text{QualityDomain}(x) \rightarrow \neg \text{QualityDimension}(x)$
a90 $\text{associatedWith}(s, t) \wedge \text{associatedWith}(s', t') \wedge t' \sqsubset t \rightarrow s' \subset s$

An intrinsic moment type is a quality type iff it is associated with a unique quality structure (a91). A *quality* is defined as an entity that instantiates a unique *quality kind* (d6).

- a91** $\text{QualityType}(t) \leftrightarrow \text{IntrinsicMomentType}(t) \wedge \exists!x(\text{QualityStructure}(x) \wedge \text{associatedWith}(x, t))$ (6.24)
d6 $\text{Quality}(x) \stackrel{def}{=} \exists!t(\text{QualityKind}(t) \wedge x :: t)$ (6.25)

The hasValue relation holds from qualities to quales (a92), where hasValue is functional (a93). From (d6), we have that a *quality* instantiates a unique *quality kind*, which from (a91) is associated to at least one quality structure. The axiom (a94) enforces each *quale* of a *quality* to be member of such *quality structures*.

- a92** $\text{hasValue}(x, y) \rightarrow \text{Quality}(x) \wedge \text{Quale}(y)$ (6.37)
a93 $\text{Quality}(x) \rightarrow \exists!y(\text{hasValue}(x, y))$ (6.39)
a94 $\text{hasValue}(x, y) \rightarrow \exists t, s(x :: t \wedge \text{associatedWith}(s, t) \wedge y \in s)$ (6.38)

UFO-A also defines what is for a quality (type) to be *simple* or *complex*. A *simple quality* is a quality that bears nothing (d7). A *complex quality* is a quality but not a simple quality (d8). A *simple quality type* is a quality type that have only simple qualities as instances (d9). Similarly, a *complex quality type* is a quality type that have only complex qualities as instances (d10). Moreover, quality dimensions can only be associatedWith simple quality types, and vice versa (a95), while quality domains can only be associatedWith complex quality types, and vice versa (a96).

- d7** $\text{SimpleQuality}(x) \stackrel{def}{=} \text{Quality}(x) \wedge \neg \exists y(\text{inheresIn}(y, x))$ (6.28)
d8 $\text{ComplexQuality}(x) \stackrel{def}{=} \text{Quality}(x) \wedge \neg \text{SimpleQuality}(x)$ (6.29)
d9 $\text{SimpleQualityType}(t) \stackrel{def}{=} \text{QualityType}(t) \wedge \forall x(x :: t \rightarrow \text{SimpleQuality}(x))$ (6.30)
d10 $\text{ComplexQualityType}(t) \stackrel{def}{=} \text{QualityType}(t) \wedge \forall x(x :: t \rightarrow \text{ComplexQuality}(x))$ (6.31)
a95 $\text{associatedWith}(x, y) \rightarrow (\text{QualityDimension}(x) \leftrightarrow \text{SimpleQualityType}(y))$ (6.34)
a96 $\text{associatedWith}(x, y) \rightarrow (\text{QualityDomain}(x) \leftrightarrow \text{ComplexQualityType}(y))$ (6.35)

Since the qualities of a complex quality $x :: X$ correspond to the quality dimensions of the quality domain associated with X , then no two distinct qualities inhering a complex quality can be of the same type (a97). Moreover, since quality dimensions are unidimensional, and quality domains can only be defined by quality dimensions, then complex qualities can only bear simple qualities (a98).

$$\mathbf{a97} \quad \text{ComplexQuality}(x) \wedge y :: Y \wedge z :: Z \wedge \text{inheresIn}(y, x) \wedge \text{inheresIn}(z, x) \wedge Y = Z \rightarrow y = z \quad (6.32)$$

$$\mathbf{a98} \quad \text{ComplexQuality}(x) \rightarrow \forall y(\text{inheresIn}(y, x) \rightarrow \text{SimpleQuality}(y)) \quad (6.33)$$

Concerning quality domains, each quality domain d associatedWith a complex quality type t can be defined in terms of the Cartesian product of the quality dimensions associatedWith the quality types characterizing t (a99).

$$\mathbf{a99} \quad \text{QualityDomain}(x) \wedge \text{associatedWith}(x, t) \rightarrow \exists y_1, \dots, y_n, z_1, \dots, z_n((x \subseteq y_1 \times \dots \times y_n) \wedge \bigwedge_{i=1}^n (\text{associatedWith}(y_i, z_i) \wedge \text{characterization}(t, z_i)) \wedge \forall w(\text{characterization}(t, w) \rightarrow \bigvee_{i=1}^n (w = z_i))) \quad (6.36)$$

A metric space s is obtained from a quality domain s by associating a distance function d (called the metric of s) such that for every two quality values x, y in s , $d(x, y)$ represents the distance between x and y in s (see Guizzardi, 2005, §6.11). Here, we define d as a left-total functional ternary relation $d(x, y, r)$ such that r is the distance between the quales x and y , see (a100), (a101).

$$\mathbf{a100} \quad d(x, y, r) \rightarrow (\text{Quale}(x) \wedge \text{Quale}(y) \wedge \exists z(\text{memberOf}(x, z) \wedge \text{memberOf}(y, z)))$$

$$\mathbf{a101} \quad (\text{Quale}(x) \wedge \text{Quale}(y)) \rightarrow \exists! r(d(x, y, r))$$

Moreover, the distance relation d must also obey the following constraints:

- a) $x = y \wedge d(x, y, r) \rightarrow (r = 0)$;
- b) $d(x, y, r) \rightarrow d(y, x, r)$;
- c) $d(x, y, r_0) \wedge d(y, z, r_1) \wedge d(x, z, r_2) \wedge +(r_0, r_1, s) \rightarrow s \geq r_2$ (triangle inequality)¹²

The notion of *region* is also informally defined by Guizzardi (2005, §6.2.6). A region is a kind of division of a quality structure with respect to a specific quale and according to the structure of the quality structure. Definition 6.12 (see quality region, Guizzardi, 2005, p.228) states that a *quality region* is a convex region c of a quality domain; while c is convex iff for all two points x, y in c , all points between x and y are also in c . We think that a suitable set theory would be able to represent the required topological notions.

3.13. Endurants and perdurants

Any endurant is connected to a perdurant by the *manifestation* relation (a102). The *life of* an endurant is then specified by a functional relation that associates the endurant with the mereological sum of all the events that manifest it (a103). The life of an endurant is unique due to the unicity of mereological sums. Moreover, the case where two perdurants are consecutive in time is here abstractly modeled by means of the *meet* relation (a104). We refer to Guizzardi et al. (2016) for the detailed presentation of this approach, and to Benevides et al. (2019) for a complete formalization of the UFO-B ontology.

$$\mathbf{a102} \quad \text{manifests}(x, y) \rightarrow \text{Endurant}(x) \wedge \text{Perdurant}(y)$$

$$\mathbf{a103} \quad \text{lifeOf}(x, y) \leftrightarrow \text{Perdurant}(x) \wedge \text{Endurant}(y) \wedge \forall z(\text{O}(z, x) \leftrightarrow \text{Perdurant}(z) \wedge \text{manifests}(z, y))$$

$$\mathbf{a104} \quad \text{meet}(x, y) \rightarrow \text{Perdurant}(x) \wedge \text{Perdurant}(y)$$

¹² $+(r_0, r_1, s)$ denotes the relationship that holds when the arithmetic equation $s = r_0 + r_1$ is true.

So, there is a special perdurant (which we may call a process, in a very particular sense) that is the life of an enduring, i.e., the sum of everything that is a manifestation of the dispositions inhering in) that perdurant (Guizzardi et al., 2016). As previously discussed, perdurants have all their parts and constituents necessarily. So, every single manifestation of that enduring literally changes its life (a change *of* life, not a change *in* the life!). In other words, this particular perdurant represents the *current life of an enduring* at each point in time (Guizzardi et al., 2016).

4. Analysis and formalization in UFO: examples

This section presents an analysis and formalization of various cases that originate from the First FOUST Workshop – the Foundational Stance, an international forum dedicated to Foundational Ontology research. They cover a number of topics (composition/constitution, roles, property change, event change, and conceptual evolution) and are also addressed by other papers in this special issue to facilitate the comparison of the various approaches. Each of the following sub-sections begins with a quote of the text supplied as the case description, which is followed by our analysis. We employ OntoUML diagrams and provide some formalization to accompany them.

4.1. Composition/constitution

- 1) “There is a four-legged table made of wood. Some time later, a leg of the table is replaced. Even later, the table is demolished so it ceases to exist although the wood is still there after the demolition.”

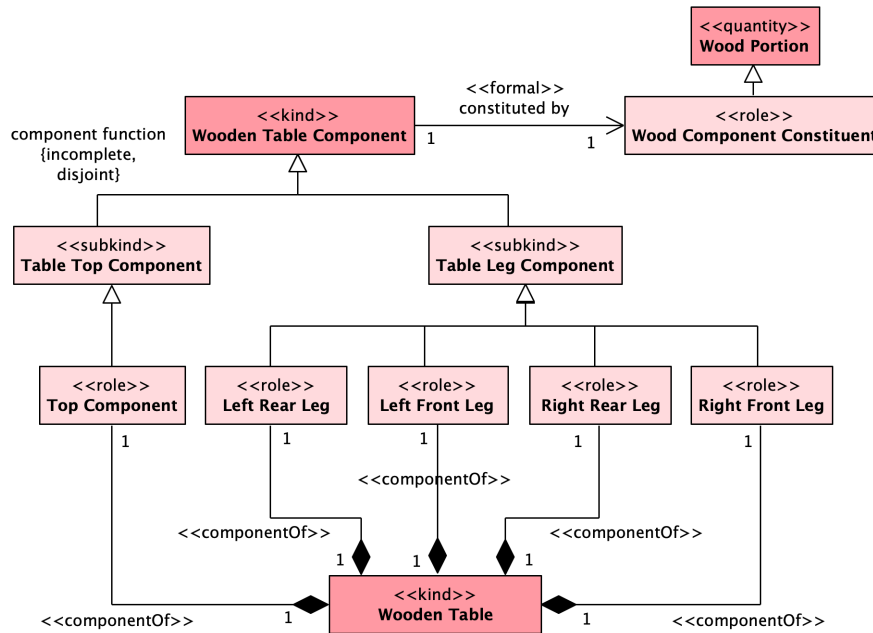


Figure 2. The Wooden Table case in OntoUML.

Figure 2 depicts an OntoUML model¹³ representing this situation. In this model, a *WoodPortion* is a quantity (in the technical sense discussed by Guizzardi (2005), i.e., a maximally topologically self-connected portion of matter). It is *contingent* for a *WoodPortion* to constitute a *Wooden Table Component*. In this model, *Wooden Table Components* are artifacts that are manufactured to necessarily serve the role of *Wooden Table Component* (i.e., *Wooden Table Component* is a nominal kind/artifact kind). Two par-

¹³In OntoUML, the stereotypes «kind» and «quantity» stand for ObjectKind and QuantityKind, respectively.

ticular subkinds considered are a **Table Leg Component** (e.g., that has necessarily a shape and necessarily a certain structure capable of supporting a certain weight range, etc.) and a **Table Top Component**. A **Wooden Table** is structured in a number of functional roles associated via **componentOf** relations, i.e., parthood relations in which the parts play a functional role with respect to the whole (see Guizzardi, 2005), that can *contingently* be played by **Wooden Table Components**. Since the relation between the table and its parts is one of *generic dependence*, parts can be replaced without any impact on the identity of the table. Moreover, since the relation between a **Wood Portion** and each **Wooden Table Component** (that it contingently constitutes) is one of constitution and not one of identity, the lifecycles of the former and the latter are completely independent. Therefore, a **Wood Portion** can survive the destruction of these wood components and the destruction of the table as a functional complex. Finally, as represented by the black diamond in the figure (exclusive parthood), each **Table Leg Component** can only be part of one single **Wooden Table** at a certain point in time.

In the sequel, we show a partial formalization of this case.

QuantityKind(**WoodPortion**)

Being a **Wooden Component Constituent** is a role played by a **Wood Portion** in the scope of a (contingent) constitution relation with a **Wooden Table Component**:

Role(**WoodenComponentConstituent**)

WoodenComponentConstituent \sqsubseteq **WoodPortion**

$\forall x.(x :: \text{WoodenComponentConstituent} \rightarrow \exists!y.(y :: \text{WoodenTableComponent} \wedge \text{constitutedBy}(y, x)))$

This constitution relation is also contingent from the point of view of the **Wooden Table Component**, i.e., the **Wooden Table Component** can be constituted by different **Wood Portions** in different situations (worlds):¹⁴

$\forall x.(x :: \text{WoodenTableComponent} \rightarrow \exists!y.(y :: \text{WoodenComponentConstituent} \wedge \text{constitutedBy}(x, y)))$

ObjectKind(**WoodenTable**)

ObjectKind(**WoodenTableComponent**)

SubKind(**TableLegComponent**)

SubKind(**TableTopComponent**)

TableTopComponent \sqsubseteq **WoodenTableComponent**

TableLegComponent \sqsubseteq **WoodenTableComponent**

To allow the representation of **Table Leg Component** and **Table Top Component** as disjoint types, we introduce (a105):

a105 $\text{isDisjointWith}(t, t') \leftrightarrow \text{Type}(t) \wedge \text{Type}(t') \wedge \neg \exists x(x :: t_1 \wedge x :: t_2)$
 $\text{isDisjointWith}(\text{TableLegComponent}, \text{TableTopComponent})$

Right Rear Leg, **Right Front Leg**, **Left Front Leg**, **Left Rear Leg** and **Top Component** are roles played by a **Table Leg Component** (the first four) and **Table Top Component** (the latter) in the scope of a *component of* relation with a **Table**:

¹⁴One should notice that these formulae here are a special case of (a58) in which there is a single constituent involved. In other words, a case of what is called *wholly constituting*. Moreover, since **WoodenTableComponent** is a kind, its instances are always generically dependent on **WoodenComponentConstituent**. In contrast, since **WoodenComponentConstituent** is a role, **WoodPortion** is only dependent on **WoodenTableComponent** insofar as it instantiates the **WoodenComponentConstituent** role.

```

Role(RightRearLeg)
Role(RightFrontLeg)
Role(LeftRearLeg)
Role(LeftFrontLeg)
Role(TopComponent)
RightRearLeg  $\sqsubseteq$  TableLegComponent
RightFrontLeg  $\sqsubseteq$  TableLegComponent
LeftRearLeg  $\sqsubseteq$  TableLegComponent
LeftFrontLeg  $\sqsubseteq$  TableLegComponent
TopComponent  $\sqsubseteq$  TableTopComponent

```

For example, Right Front Leg is a role played by a Table Leg Component when being a functional part of a Wooden Table. Guizzardi (2005) defines the relation of *component of* between a functional part and a functional complex, and it conflates a mereological relation plus a relation of functional dependence. Here, in order for a Table Component to function as a Table Component, it must be part of a Table functioning as a Table (*Mutatis mutandis*, the same for the roles of Right Rear Leg, Left Front Leg, Left Rear Leg and Top Component):

$$\forall x.(x :: \text{RightFrontLeg} \rightarrow \exists!y.(y :: \text{WoodenTable} \wedge \text{componentOf}(x, \text{RightFrontLeg}, y, \text{WoodenTable})))$$

In the converse direction, in order for a Wooden Table to function as such it must be composed of four distinct Table Leg Components and a TopComponent (which here are all assumed to be Wooden Table Components), each of these functioning in a particular role:

$$\begin{aligned} \forall x.(x :: \text{WoodenTable} \rightarrow \exists y_1, y_2, y_3, y_4, y_5. & (y_1 :: \text{RightRearLeg} \wedge y_2 :: \text{RightFrontLeg} \wedge \\ & y_3 :: \text{LeftFrontLeg} \wedge y_4 :: \text{LeftRearLeg} \wedge y_5 :: \text{TopComponent} \wedge \\ & \text{componentOf}(y_1, \text{RightRearLeg}, x, \text{WoodenTable}) \wedge \\ & \text{componentOf}(y_2, \text{RightFrontLeg}, x, \text{WoodenTable}) \wedge \\ & \text{componentOf}(y_3, \text{LeftFrontLeg}, x, \text{WoodenTable}) \wedge \\ & \text{componentOf}(y_4, \text{LeftRearLeg}, x, \text{WoodenTable}) \wedge \\ & \text{componentOf}(y_5, \text{TopComponent}, x, \text{WoodenTable}) \wedge \\ & (y_1 \neq y_2) \wedge (y_1 \neq y_3) \wedge (y_1 \neq y_4) \wedge (y_2 \neq y_3) \wedge (y_2 \neq y_4) \wedge (y_3 \neq y_4)) \end{aligned}$$

The relation of *component of* is a relation of generic dependence from the whole to the part and from the part to the whole. So, not only the Wooden Table can have different Wooden Table Components playing these functional part roles in different situations, but these components can play these roles in different tables in different situations.

From an OntoUML model and using its supporting tools, we can automatically generate models that satisfy the logical theory corresponding to the OntoUML model (Benevides et al., 2010). In conceptual modeling terms, this means that we can generate instances that are admissible by an OntoUML model. In the sequel, we show a number of examples (Figures 3 to 5) of these visual models automatically generated for the OntoUML diagram of Figure 2.

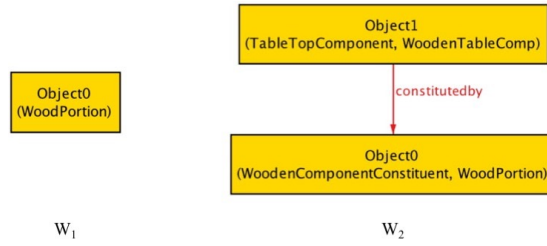


Figure 3. A WoodPortion exists in world w_1 without constituting any object (Object0::WoodPortion) and in w_2 it goes on to constitute a Wooden Table Component, namely, a Table Top Component (Object1::WoodenTableComponent, Object1::TableTopComponent).



Figure 4. A Table Leg Component (Object3::TableLegComponent, Object3::WoodenTableComponent) exists in w_1 being constituted by a Wood Portion (Object7::WoodPortion) and, in w_2 , it still exists but now constituted by a different Wood Portion (Object4::WoodPortion).

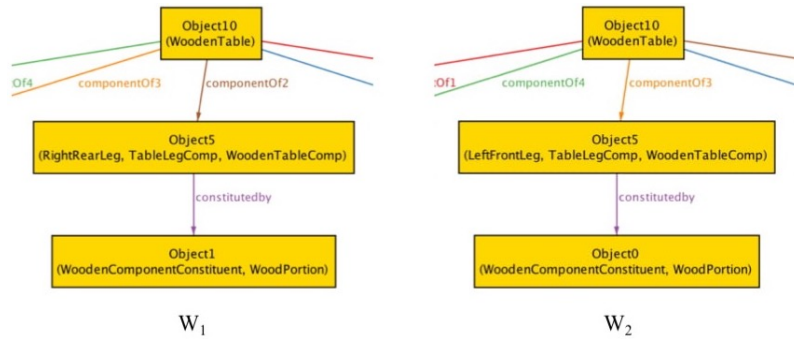


Figure 5. A WoodenTable (Object10::WoodenTable) exists in world w_1 being composed by a Table Leg Component that plays in it a functional role of Right Rear Leg (Object5::WoodenTableComponent, Object5::TableLegComponent, Object5::RightRearLeg, componentOf(Object5, RightRearLeg, Object10, WoodenTable)) and which, in that world, is constituted by a particular WoodPortion (Object1::WoodPortion, Object1::WoodenComponentConstituent, constitutedBy(Object5, Object1)). In a different world w_2 , the table still exists but now having that same Table Leg Component Object5 playing a different role, namely, that of a Left Front Leg (Object5::LeftFrontLeg, componentOf(Object5, LeftFrontLeg, Object10, WoodenTable)). In that world, this Table Leg Component is also constituted by a different Wood Portion (constitutedBy(Object5, Object0)).

4.2. Roles

- 2) “Mr. Potter is the teacher of class 2C at Shapism School and resigns at the beginning of the spring break. After the spring break, Mrs. Bumblebee replaces Mr. Potter as the teacher of 2C. Also, student Mary left the class at the beginning of the break and a new student, John, joins in when the break ends.”

In this example, captured in the OntoUML of Figure 6, Teacher is a role played by a person when mediated by an Employment relator (Guarino and Guizzardi, 2015) connecting her to a School. Analogously, Student is a role played by a Person when connected via a School Enrollment relator to a School. A School makes Course Offerings (a relator) to a given Class (a collective forming the target community

to which that course is addressed). Students can enroll to a Course Offering via a Course Enrollment relator, connecting the former to a particular enrollment of a Student in that School. In other words, the enrollment of a Student to a Course Offering *existentially depends* both on the offering and on a particular School Enrollment of that Student. Notice that the relation from Course Enrollment to Course Offering is one of existential dependence (mediation, Guizzardi, 2005; Guarino and Guizzardi, 2015) and, hence, immutable. However, the relation from Course Offering to Course Enrollment is an optional relation of generic dependence and, thus, contingent, mutable. For this reason, the set of students actually enrolled in a given Course Offering can change from situation to situation. Analogously, a Teacher is assigned to a Course Offering by having a Course Teacher Assignment relator that connects a Course Offering to an Employment of a Teacher. In other words, a teacher's assignment to a course offering existentially depends on both on the Course Offering and on the Employment of a Teacher in that School. Again, since the relation from Course Offering to Teacher Assignment is one of generic dependence, different teachers can be assigned to a course offering in different points in time.

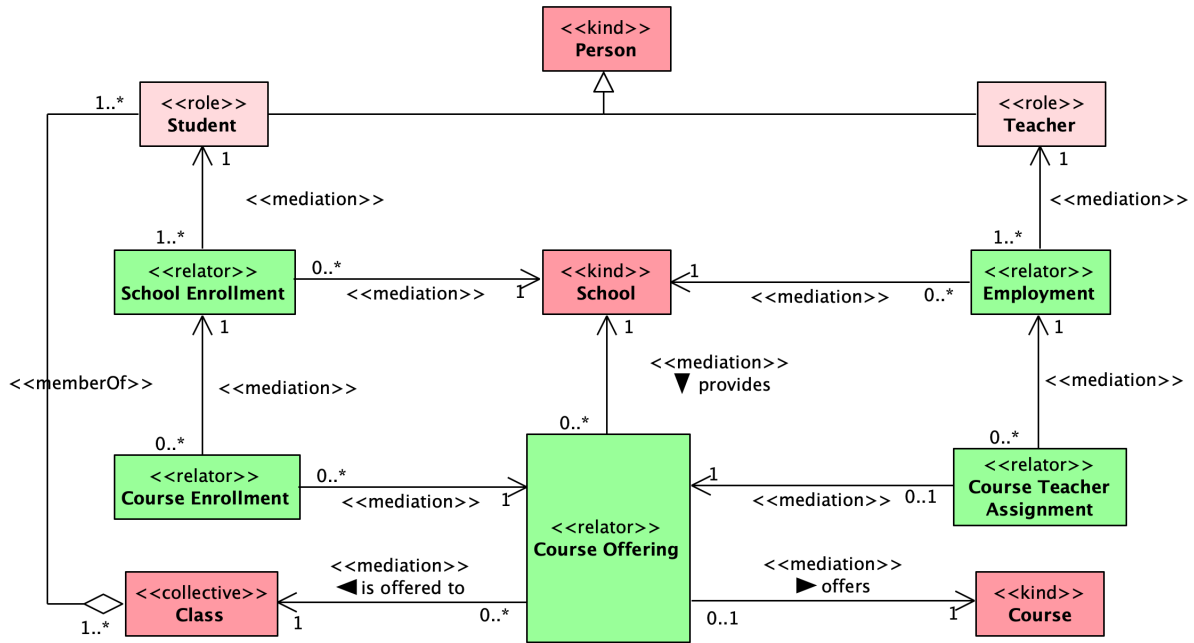


Figure 6. The School case in OntoUML.

In the sequel, we show a partial formalization of this case.

```
ObjectKind(School)
ObjectKind(Course)
ObjectKind(Person)
CollectiveKind(Class)
```

Teacher is a role (contingently) played by a Person in the scope of an Employment relation to a School.

```
Role(Teacher)
Teacher ⊆ Person
RelatorKind(Employment)
 $\forall x(x :: \text{Teacher} \rightarrow \exists y(y :: \text{Employment} \wedge \text{mediates}(y, x))$ 
```

$$\forall x(x :: \text{Employment} \rightarrow \exists !y, !z(y :: \text{Teacher} \wedge z :: \text{School} \wedge \text{mediates}(x, y) \wedge \text{mediates}(x, z)))$$

Student is a role (contingently) played by a Person in the scope of an Enrollment relation to a School.

Role(Student)

Student \sqsubseteq Person

RelatorKind(SchoolEnrollment)

$$\forall x(x :: \text{Student} \rightarrow \exists y(y :: \text{SchoolEnrollment} \wedge \text{mediates}(y, x))$$

$$\forall x(x :: \text{SchoolEnrollment} \rightarrow \exists !y, !z(y :: \text{Student} \wedge z :: \text{School} \wedge \text{mediates}(x, y) \wedge \text{mediates}(x, z)))$$

A School can make several Course Offerings to Classes. Students can be members of Classes. A Class must have at least two Students as members. Notice that a Course Offering in this model is a particular instantiation of the UFO-S Service Offering pattern (Nardi et al., 2015), having the Classhere as the Target Community.

RelatorKind(CourseOffering)

$$\forall x(x :: \text{CourseOffering} \rightarrow \exists !y, !z, w(y :: \text{School} \wedge z :: \text{Course} \wedge w :: \text{Class} \wedge \text{mediates}(x, y) \wedge \text{mediates}(x, z) \wedge \text{mediates}(x, w)))$$

$$\forall x(x :: \text{Student} \rightarrow \exists y(y :: \text{Class} \wedge \text{memberOf}(x, y))$$

$$\forall x(x :: \text{Class} \rightarrow \exists y, z(y :: \text{Student} \wedge z :: \text{Student} \wedge (y \neq z) \wedge \text{memberOf}(y, x) \wedge \text{memberOf}(z, x)))$$

Students that are enrolled in a School can enroll in Course Offerings of that School. Notice that a Course Enrollment here is an instantiation of the UFO-S pattern of Service Agreements (Nardi et al., 2015).¹⁵

RelatorKind(CourseEnrollment)

$$\forall x(x :: \text{CourseEnrollment} \rightarrow \exists !y, !z(y :: \text{SchoolEnrollment} \wedge z :: \text{CourseOffering} \wedge \text{mediates}(x, y) \wedge \text{mediates}(x, z)))$$

Teachers can be assigned to Course Offerings. There is exactly one Teacher assigned to a Course Offering in a given situation. However, since the relation between Course Offering and Course Teacher Assignment is one of generic dependence, different Teachers can be assigned to the same Course Offering in different situations.¹⁶

RelatorKind(CourseTeacherAssignment)

$$\forall x(x :: \text{CourseTeacherAssignment} \rightarrow \exists !y, !z(y :: \text{Employment} \wedge z :: \text{CourseOffering} \wedge \text{mediates}(x, y) \wedge \text{mediates}(x, z)))$$

$$\forall x, y, z(x :: \text{CourseOffering} \wedge y :: \text{CourseTeacherAssignment} \wedge z :: \text{CourseTeacherAssignment} \wedge \text{mediates}(y, x) \wedge \text{mediates}(z, x) \rightarrow (y = z))$$

Figure 7 shows visual models automatically generated for the OntoUML diagram of Figure 6 illustrating the dynamics of change in this domain.

¹⁵The complete specification of this case includes a constraint that guarantees that a Student can only enroll in Course Offerings of the Schools in which he has a School Enrollment. This constraint is automatically detected and included in the specification by the anti-pattern detection and rectification support of the OntoUML tool (Guerson et al., 2015). We omit it here, nonetheless, for the sake of brevity.

¹⁶The complete specification of this case includes a constraint that guarantees that a Teacher can only be assigned to Course Offerings of the Schools in which she has an Employment relationship. Once more, this constraint is automatically detected and included in the specification by the anti-pattern detection and rectification support of the OntoUML tool. Once more, we omit it here, nonetheless, for the sake of brevity.

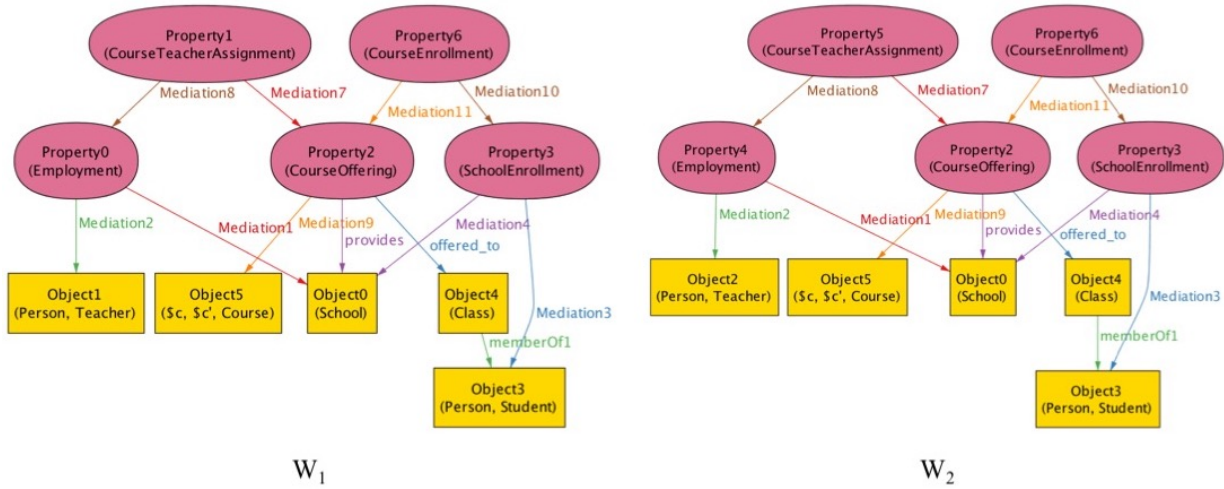


Figure 7. In world w_1 , we have a School Object0 (Object0::School) and a student Object3 (Object3::Student) enrolled in that School (by Property3 (Property3::SchoolEnrollment)) and who is part of Class Object4 (Object4::Class) that works for that School (by Employment Property0 (Property0::Employment)). There is also a Course Offering (Property2::CourseOffering) of Course Object5 (Object5::Course, e.g., CS 101) made by that School to Class Object4. Finally, Student Object3 enrolls in Course Offering Property2 (by Course Enrollment Property6 (Property6::CourseEnrollment)) and Teacher Object1 is assigned to teach that Course Offering (by Course Teacher Assignment Property1 (Property1::CourseTeacherAssignment)). In world w_2 , a different Teacher Object2 (Object2::School), who is also employed in that same school (by Employment Property4 (Property4::Employment)), is now assigned to that Course Offering Property2.

4.3. Property change

3.a) “A flower is red in the summer. As time passes, the color changes. In autumn the flower is brown.”

In Figure 8, we present an OntoUML model¹⁷ representing this situation. In this model, a Rose is modeled as a subkind of Flower. All flowers are bearers of a Color quality¹⁸. As for any quality, a Flower Color¹⁹ quality inheres in its bearer (the characterization relation). A perceivable quality takes its value in a color domain (here modeled as a simple ordinal space of colors).

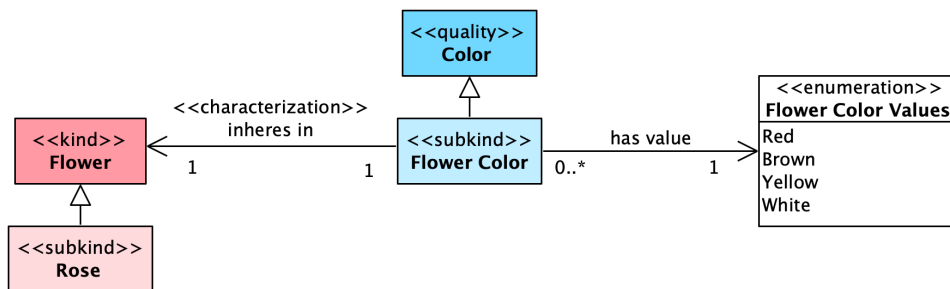


Figure 8. The Flower case in OntoUML.

In the sequel, we show a partial formalization of this case.

¹⁷In OntoUML, the simplest way to represent quality structures is as datatypes (Guizzardi, 2005). Classes stereotyped as «enumeration» are particular types of datatypes. For a more sophisticated representation of these structures in the language, one should refer to Albuquerque and Guizzardi (2013). We take here the simplest approach.

¹⁸To be precise, Color is a particular type of quality termed as Perceivable Quality by Albuquerque and Guizzardi (2013). We omit this discussion here.

¹⁹Flower Color is a genuine subtype of Color given that it is associated with a particular subspace of possible values, which is a proper part of the entire color spindle.

ObjectKind(Flower)
 SubKind(Rose)
 Rose \sqsubseteq Flower
 SubKind(Color)
 SubKind(FlowerColor)
 FlowerColor \sqsubseteq Color
 QualityStructure(FlowerColorValues)

Now, the quality type Flower Color characterizes the object type Flower:

characterization(Flower, FlowerColor)

Following (a81) and (a82), this implies:

$$\forall x.(x :: \text{Flower} \rightarrow \exists! y.(y :: \text{FlowerColor} \wedge \text{inheresIn}(y, x)))$$

$$\forall z.(z :: \text{FlowerColor} \rightarrow \exists! w.(w :: \text{Flower} \wedge \text{inheresIn}(z, w)))$$

Since Flower Color is a Quality Type, it is associated with a Quality Structure (a91):

associatedWith(FlowerColor, FlowerColorValues)

As Quality Structures are non-empty sets of possible values that qualities can take (a86):

$$\text{Red} \in \text{FlowerColorValues} \wedge \text{Yellow} \in \text{FlowerColorValues} \wedge$$

$$\text{Brown} \in \text{FlowerColorValues} \wedge \text{White} \in \text{FlowerColorValues}$$

Given (a93) and (a94), each quality instance of Flower Color takes, in a given situation, a value in the Flower Color Values space:

$$\forall x.(x :: \text{FlowerColor} \rightarrow \exists! y.(y \in \text{FlowerColorValues} \wedge \text{hasValue}(x, y)))$$

Now, notice that inherence is an existential dependence relation and, hence, the connection between a particular quality and its bearer is immutable. However, the relation of hasValue between a quality and its value (qualia) in a quality structure is one of generic dependence. Therefore, the value of a given Flower Color can change from situation to situation. In the sequel, we show examples (Figure 9) of the visual models automatically generated for the OntoUML diagram of Figure 8.

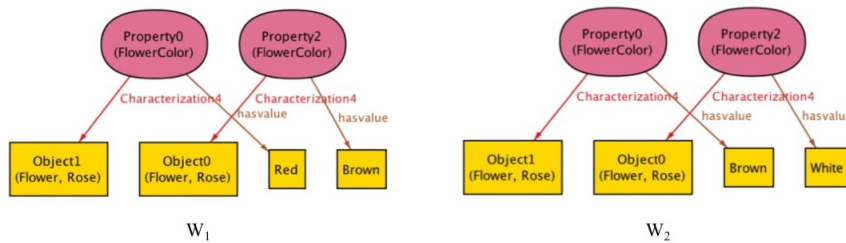


Figure 9. In world w_1 , we have two roses Object0 (Object0::Rose, Object0::Flower) and Object1 (Object1::Rose, Object1::Flower). These roses bear each a Flower Color quality (Property0::FlowerColor, inheresIn(Property0, Object1), Property2::FlowerColor, inheresIn(Property2, Object0)). In that situation, qualities Property0 and Property2 have the color values Red and Brown, respectively. In world w_2 , Rose Object1 turns from Red to Brown (by having its quality Property0 assuming the new value), and Rose Object0 turns from Brown to White (again, by having its quality Property2 taking the new value).

3.b) “A man is walking when suddenly he starts walking faster and then breaks into a run.”

In Figure 10, we present an OntoUML model representing this case. In UFO, we follow a classical view of events in which events are *modally fragile* entities. So, events cannot bear modal properties, they cannot genuinely change, and they cannot be different from what they are (Guizzardi et al., 2016). Events are also polygenic manifestations of (possibly bundles of) dispositions (Guizzardi et al., 2013). These (bundles of) dispositions are said to be the focuses of these events, so that events are carved out of *scenes* by having those underlying durants as their focus (Guarino and Guizzardi, 2016). For example, the marriage of John and Mary as a perdurant is the manifestation of the marriage (as a relator) that binds John and Mary, i.e., the sum of all manifestations of the moments inhering in John-qua-Husband-of-Mary and Mary-qua-Wife-of-John (Guizzardi et al., 2016, 2013). Following up on this example, there are two kinds of “changes” referring to “John and Mary’s marriage”: (1) a *change in* the marriage relator—for example, when we say that the marriage between John and Mary (as a whole) changed from passionate in situation s_1 to cold and distant in situation s_2 , we are referring to this enduring (a bundle of moments), which can qualitatively change while remaining numerically the same; (2) a *change of* (or, more precisely, a variation of) temporal parts of a perdurant that are qualitatively different. Figure 10 illustrates a case of the latter; Figure 12 illustrates a case of the former.

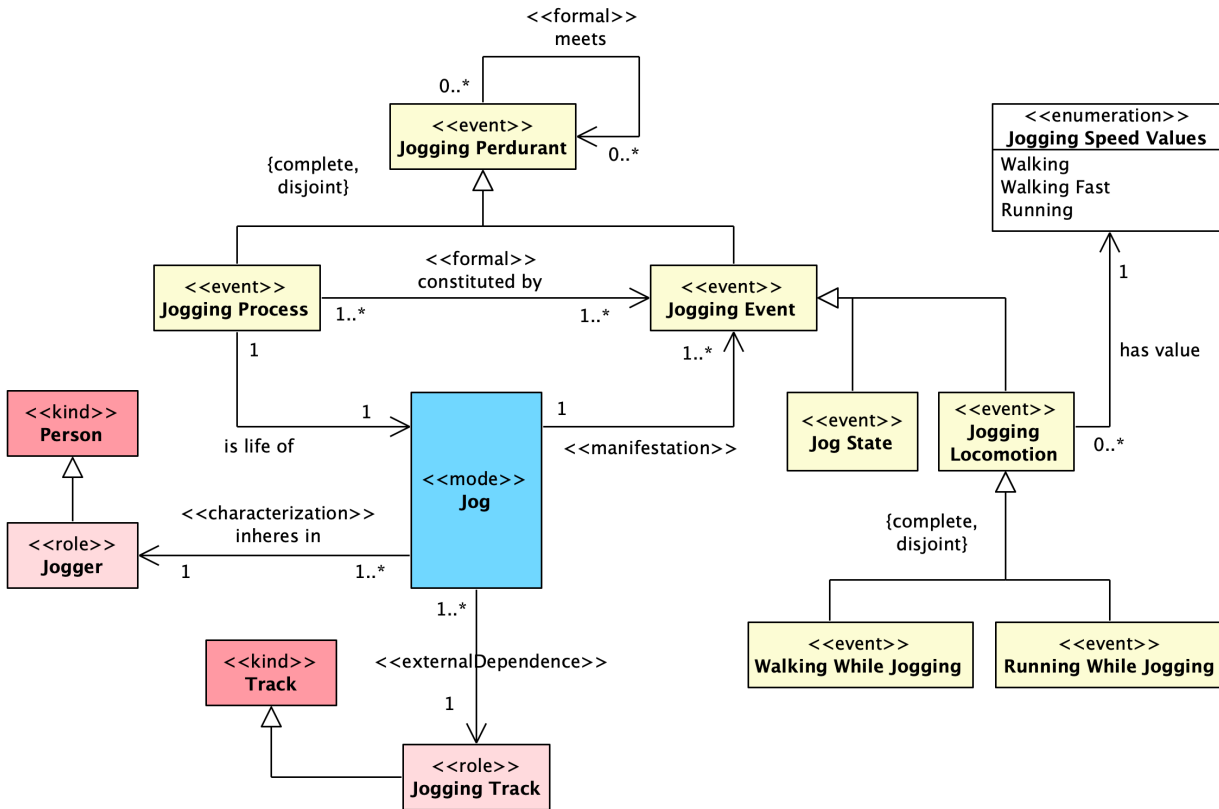


Figure 10. The Changing Speed case in OntoUML.

In the sequel, we show a partial formalization of this case.

ObjectKind(Person)

Jogger is a role played by a Person when bearing a Jog mode (an enduring comprising an intention as well as the capacities of the person as a Jogger).

Role(Jogger)

Jogger \sqsubseteq Person

ModeKind(Jog)

characterization(Jogger, Jog)

$$\forall x.(x :: \text{Jog} \rightarrow \exists!y.(y :: \text{JoggingTrack} \wedge \text{externallyDependent}(x, y)))$$

As explained in details by Guizzardi et al. (2016), associated to a Jog (an endurant), we have Jogging perdurants that are manifestations of that endurant. In particular, we have Jogging Events, which are direct manifestations of the dispositions constituting a Jog, and Jogging Processes, which are constituted by Jogging Events.

EventType(JoggingProcess)

EventType(JoggingEvent)

$$\forall x.(x :: \text{JoggingEvent} \rightarrow \exists!y.(y :: \text{Jog} \wedge \Box(\text{ex}(x) \rightarrow \text{manifestedBy}(y, x))))$$

At each situation, we have a particular type of maximal Jogging Process (termed the life of a Jog) that is constituted by (exactly) the sum of Jogging Events that are manifestations of that Jog. Moreover, since perdurants have all their parts, properties and constituents necessarily, every time a new Jogging Event takes place, there is a new Jogging Process that arises as a cumulative aggregation of manifestations of that Jog, i.e., a new life of that Jog.

$$\forall x.(x :: \text{Jog} \rightarrow \exists!y.(y :: \text{JoggingProcess} \wedge \text{lifeOf}(y, x))$$

$$\forall x.(x :: \text{JoggingProcess} \rightarrow \exists y.(y :: \text{JoggingEvent} \wedge \text{constitutedBy}(x, y))$$

$$\forall x, y.(x :: \text{Jog} \wedge y :: \text{JoggingProcess} \wedge \text{lifeOf}(y, x) \rightarrow \forall z.(z :: \text{JoggingEvent} \rightarrow (\text{constitutedBy}(y, z) \leftrightarrow \text{manifestedBy}(x, z))))$$

Since perdurants unfold in time accumulating parts, if we have two Jogging Processes that are manifestations of the same Jog and that immediately follow each other, *meeting* in the Allen sense (Guizzardi et al., 2013), then we have that all constituents of the preceding process also constitute the succeeding process²⁰.

$$\forall x, y, z.(x :: \text{Jog} \wedge y :: \text{JoggingProcess} \wedge z :: \text{JoggingProcess} \wedge \text{manifestedBy}(x, y) \wedge \text{manifestedBy}(x, z) \wedge \text{meet}(y, z) \rightarrow \forall w.(w :: \text{JoggingEvent} \wedge \text{constitutedBy}(y, w) \rightarrow \text{constitutedBy}(z, w)))$$

Jogging Events can be stative/homeomerous and some are dynamic/sequences of changes. In the former's case, we have Jog State, while in the latter we have Jogging Locomotion. Moreover, Jogging Locomotions can be such that they have the quality of being a Walk While Jogging event (an event characterized by a “walking” moving speed), or they can be such that they have the quality of being a Run While Jogging (an event characterized by a “running” moving speed). Since events are immutable, they are characterized by tropes, in the classical sense, not qualities as *variable tropes* (Guarino and Guizzardi, 2015, 2016). Unlike qualities, tropes cannot change their values.

EventType(JogState)

EventType(JoggingLocomotion)

EventType(WalkWhileJogging)

EventType(RunWhileJogging)

$$\forall x.(x :: \text{JoggingLocomotion} \rightarrow \exists!y.(y \in \text{JoggingSpeedValues} \wedge \Box \text{hasValue}(x, y)))$$

In the specialization of Jogging Event into Jogging Locomotion and Jog State, and the specialization Jogging Locomotion into Walk While Jogging and Run While Jogging, partition the instances of the specialized types, which can be represented with support of (a105)–(a107).

²⁰We use the term *process* here in this very specific sense to denote the current life of an endurant constituted by all and exactly the manifestations of (the dispositions inhering in) that endurant (see section 3.13)(Guizzardi et al., 2016). These formulae here capture the continuous changes of life of an endurant as dispositions continue to manifest. Each current life of an endurant is monotonically constituted by all constituents of its immediate preceding life.

- a106** $\text{isCompletelyCoveredBy}(t, t', t'') \leftrightarrow \forall x(x :: t \rightarrow x :: t' \vee x :: t'')$
a107 $\text{isPartitionedInto}(t, t', t'') \leftrightarrow \text{isCompletelyCoveredBy}(t, t', t'') \wedge \text{isDisjointWith}(t', t'')$
 $\text{isPartitionedInto}(\text{JoggingEvent}, \text{JogState}, \text{JoggingLocomotion})$
 $\text{isPartitionedInto}(\text{JoggingLocomotion}, \text{WalkWhileJogging}, \text{RunWhileJogging})$

Figure 11 depicts instances automatically generated for the OntoUML diagram of Figure 10.

4.4. Event change

- 4) “A man is walking to the station, but before he gets there, he turns around and goes home.”

As discussed in the previous sections, in UFO, we take the classical approach towards events, thus, treating them as modally fragile entities. For this reason, events cannot genuinely change, i.e., they cannot qualitatively change as a whole while maintaining their numerical identity. Alleged events change are either: (a) event variation — different temporal parts of the event in question have different properties as a

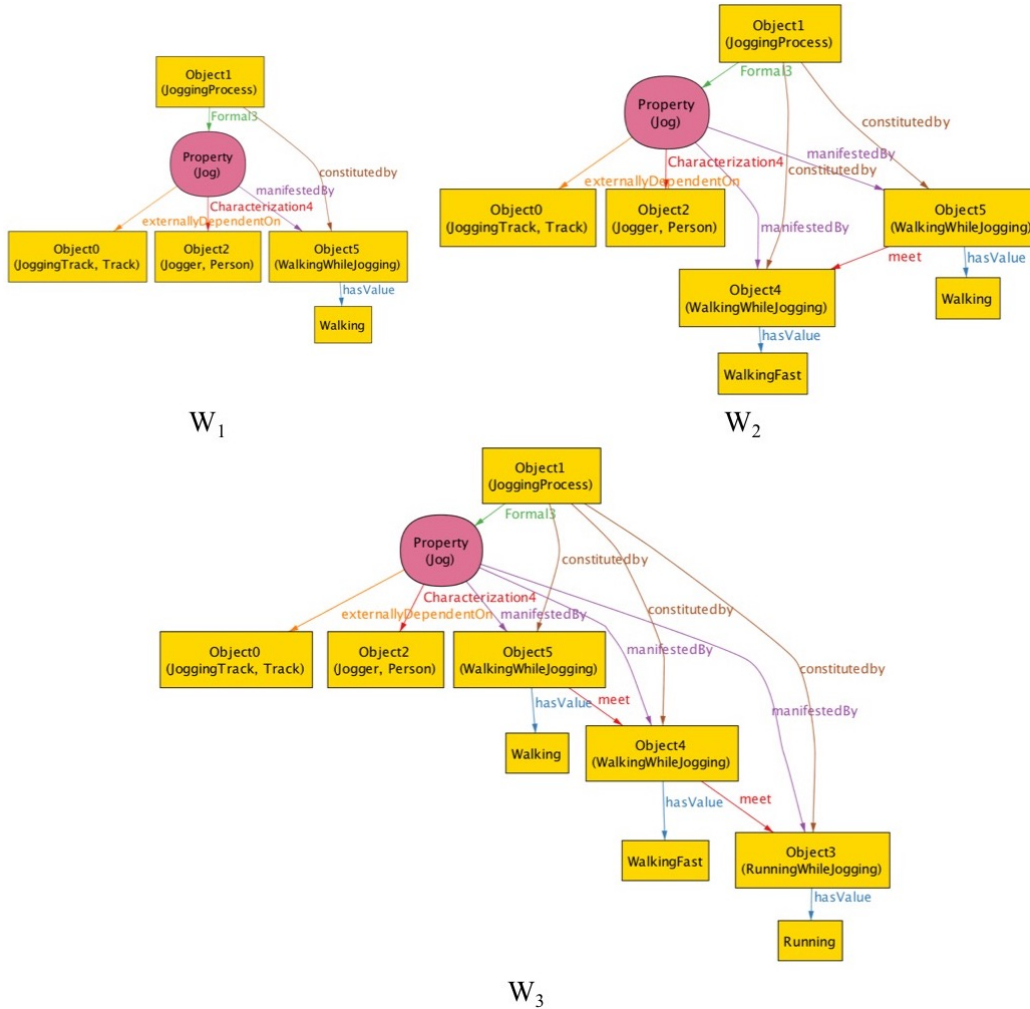


Figure 11. From world w_1 to w_3 , we have a jogger Object2 (Object2::Jogger) who is an increasing speed jog (Property::Jog, inheresIn(Property, Object2)). In w_1 , the jog is manifested by a walking speed jog Object5 (Object5::WalkingWhileJogging), which is then manifested by a fast walking Object4 (Object4::WalkingWhileJogging) in w_2 , and finally, breaks into a run Object3 (Object3::RunningWhileJogging) in w_3 . These events meet as they succeed one another being constituents of the same jogging process Object1 (Object1::JoggingProcess, meet(Object5, Object4), meet(Object4, Object3), constitutedBy(Object1, Object5), constitutedBy(Object1, Object4), constitutedBy(Object1, Object3)).

whole. An example of this is exercised in the previous section; (b) the subject of change is the focus of the event, i.e., the underlying enduring of which the event is a manifestation. The case addressed in this section is not really a case of the former but one of the latter. To put it baldly, what genuinely changes is this case is the complex mode (including an intentional component) inhering in the Walker. Due to a change in the intention of the walker, a different walking perdurant will be manifested. In other words, what genuinely changes is a property of Walk (the complex mode). These changes make that Walk be manifested as a (numerically) different walking perdurant. Given that the focus of change here is the enduring, we will refrain from representing the perdurant part of this example, which would otherwise follow the design pattern proposed by Guizzardi et al. (2016) and instantiated in the previous section.

In Figure 12, we present an OntoUML model representing this situation.²¹ In this model, a Walk is modeled as an externally dependent mode that inheres in the Walker and that is externally dependent on Place (that plays the role of an originally intended destination).

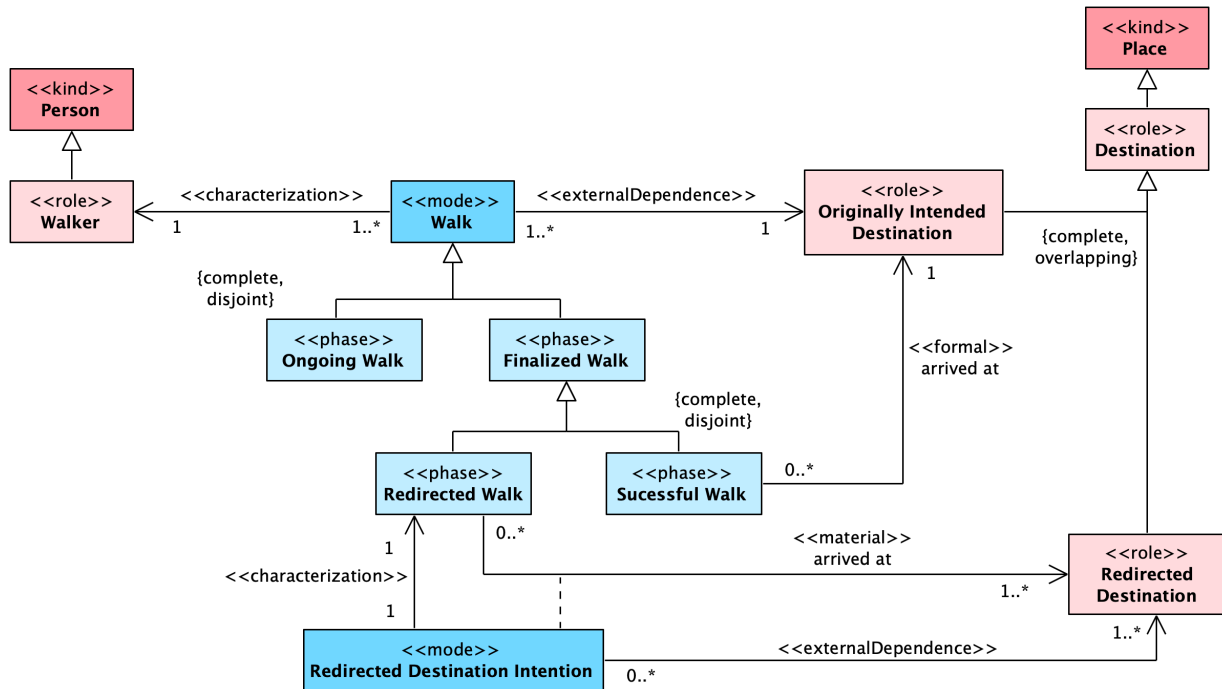


Figure 12. The Redirected Walk case in OntoUML.

ObjectKind(Person)

Role(Walker)

Walker \sqsubseteq Person

ModeKind(Walk)

characterization(Walk, Walker)

$\forall x.(x :: \text{Walk} \rightarrow \exists! y.(y :: \text{OriginallyIntendedDestination} \wedge \text{externalDependence}(x, y)))$

A (originally intended) destination is a role played by a Place when involved in a relationship with a Walk (i.e., when it is the target of an intention present in a Walk).

ObjectKind(Place)

Role(Destination)

²¹It would, of course, be trivial to model the class Man here as a subkind of Person and as a supertype of Walker. We omit it, however, for the sake of simplicity and given that it does not have any impact on the focus of the analysis conducted here.

Role(OriginallyIntendedDestination)

Destination \sqsubseteq Place

OriginallyIntendedDestination \sqsubseteq Destination

$\forall x.(x :: \text{OriginallyIntendedDestination} \rightarrow \exists y.(y :: \text{Walk} \wedge \text{externalDependence}(y, x)))$

As previously discussed, a Walk here is an endurant that is the *focus* of a walking perdurant, in the technical sense discussed by Guarino and Guizzardi (2016). In this sense, the Walk aggregates both the intention of the walker and his capabilities qua-walker. As such, i.e., as an endurant, the Walk can change in time undergoing the phases of OngoingWalk and FinalizedWalk (Guarino and Guizzardi, 2016).

isPartitionedInto(Walk, OngoingWalk, FinalizedWalk)

$\forall x.(x :: \text{FinalizedWalk} \rightarrow \exists!y.(y :: \text{Destination} \wedge \text{arrivedAt}(x, y)^{22}))$

A FinalizedWalk is only contingently a SuccessfulWalk (i.e., a walk that arrives at the originally intended destination) as well as it can be contingently (i.e., in a counterfactual situation) a RedirectedWalk.

isPartitionedInto(FinalizedWalk, SuccessfulWalk, RedirectedWalk)

$\forall x.(x :: \text{SuccessfulWalk} \leftrightarrow (x :: \text{FinalizedWalk} \wedge \forall y, z.((y :: \text{OriginallyIntendedDestination} \wedge \text{externallyDependent}(x, y) \wedge z :: \text{Destination} \wedge \text{arrivedAt}(x, z)) \rightarrow (y = z))))^{23}$

In the case of a RedirectedWalk, there is a new intention, which is aggregated to the original walk and which is directed (again, externally dependent on) a new destination. Associated to the Walk (i.e., the endurant), there are many possible unfoldings (in different histories of the world) representing how the dispositions aggregated in the walk mode are manifested by different walking perdurants (Guizzardi et al., 2016).

Role(RedirectedDestination)

RedirectedDestination \sqsubseteq Destination

ModeKind(RedirectedDestinationIntention)

$\forall x.(x :: \text{RedirectedDestinationIntention} \rightarrow \exists!y.(y :: \text{RedirectedWalk} \wedge \text{inheresIn}(x, y)))$

$\forall x.(x :: \text{RedirectedWalk} \rightarrow \exists!y.(y :: \text{RedirectedDestinationIntention} \wedge \text{inheresIn}(y, x)))$

$\forall x.(x :: \text{RedirectedDestinationIntention} \rightarrow \exists!y.(y :: \text{RedirectedDestination} \wedge \text{externallyDependent}(x, y)))$

$\forall x.(x :: \text{RedirectedDestination} \rightarrow \exists y.(y :: \text{RedirectedDestinationIntention} \wedge \text{externallyDependent}(x, y))$

$\forall x.(x :: \text{RedirectedWalk} \rightarrow \forall y, z.((y :: \text{OriginallyIntendedDestination} \wedge \text{externallyDependent}(x, y) \wedge z :: \text{RedirectedDestination} \wedge \text{arrivedAt}(x, z)) \rightarrow (y \neq z)))^{24}$

$\forall x, y, z.((x :: \text{RedirectedWalk} \wedge y :: \text{RedirectedDestinationIntention} \wedge \text{inheresIn}(y, x) \wedge z :: \text{RedirectedDestination} \wedge \text{externallyDependent}(y, z)) \rightarrow \text{arrivedAt}(x, z))$

In the sequel, we show an example of visual model (Figure 13) automatically generated for the OntoUML diagram of Figure 12.

²²The relation of “arrived at” is a material relation, which is derived from the following rule: a walk x , qua-endurant, is related via this relation to a place y iff y is the location of the Walker in the post-situation immediately brought about by the occurrence of the walking process that is the manifestation of x (Guizzardi et al., 2016).

²³A successful walk must “arrive at” a place that is identical to the originally intended destination. This constraint is automatically generated by the OntoUML tool after an automated process of anti-pattern detection and rectification (Sales and Guizzardi, 2015).

²⁴A redirected walk must “arrive at” a place that is different from the originally intended destination. Again, this constraint is automatically generated by the OntoUML tool after an automated process of anti-pattern detection and rectification.

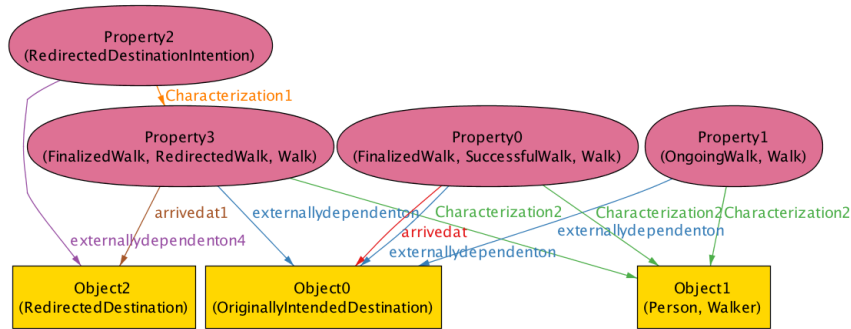


Figure 13. In this model, we have a Person Object1 that plays the role of a Walker by being the bearer of three distinct Walk modes (Object1::Person, Object1::Walker, Property0::Walk, Property1::Walk, Property2::Walk, inheresIn(Property0, Object1), inheresIn(Property1, Object1), inheresIn(Property3, Object1)). Property1 is an ongoing walk that has Object0 as its originally intended destination (Property1::OngoingWalk, Object0::Place, Object0::Destination, externallyDependent(Property1, Object0), Object0::OriginallyIntendedDestination); Property0 is a finalized walk that also had Object1 as its originally dependent and also final destination (externallyDependent(Property0, Object0), arrivedAt(Property0, Object0)); finally, Property3 is a finalized walk that had Object0 as its originally dependent but it was redirected and arrived at a different destination (externallyDependent(Property3, Object0), arrivedAt(Property3, Object2)).

4.5. Concept evolution

- 5) “A marriage is a contract that is regulated by civil and social constraints. These constraints can change but the meaning of marriage continues over time.”

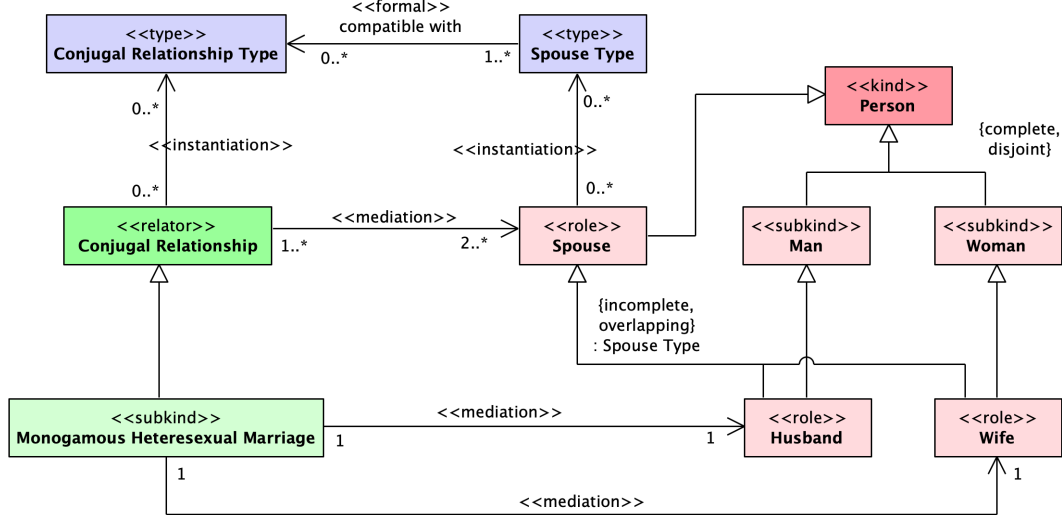


Figure 14. The Concept Evolution case in OntoUML.

In order to address matters of “concept evolution,” it is key to understand what the subject of “evolution” is, and what aspects of this subject remain stable. According to the “marriage” concept evolution case, “a marriage is a contract that is regulated by civil and social constraints”, and “these constraints can change but the meaning of marriage continues over time.” We observe that what remains stable here is a general kind of relator we call Conjugal Relationship between Spouses, as represented in Figure 14. Varying constraints established in the social process in time lead to specific types of conjugal relationships, e.g., “monogamous marriage,” “polygamous marriage,” “heterosexual marriage,” and “same-sex marriage.” What is invariant across all foreseen types of conjugal relationships is that they specialize the more general Conjugal Relationship relator kind. However, the number and sex of spouses may vary according to the specific type of conjugal relationship.

UFO captures this scenario by accounting not only for the individuals in the domain of inquiry but types as well. This leads to higher-order types, i.e., types whose instances are themselves types²⁵. The theory provides a formal characterization of the relations between higher-order types, first-order types, and individuals, along with a characterization of the so-called powertype pattern in the conceptual modeling literature). In this case, we have that specializations of the more general (and stable) notion of `ConjugalRelationship` are instances of a second-order type `ConjugalRelationshipType`. These instances establish more specific constraints that are not required in every `ConjugalRelationship`. The type `ConjugalRelationship` serves as the so-called base type in its relation with the higher-order type (Carvalho and Almeida, 2018). Identifying the second-order type allows us to specify invariants that apply to any first-order type instantiating the second-order type: First, all its instances specialize `ConjugalRelationship` (this is what remains the same or is said to continue over time). Further, by identifying the second-order type, we can impose conditions on the first-order types that instantiate it, such as, e.g., that any instance of `ConjugalRelationshipType` determines one or more specific `SpouseTypes`. For a `MonogamousHeterosexualMarriage`, for example, these are `Husband` and `Wife`. The formal relation between these higher-order types and their base types is captured with (a108), as defined by Carvalho and Almeida (2018) for MLT.

a108 $\text{categorizes}(t_1, t_2) \leftrightarrow \text{Type}(t_1) \wedge \forall t_3 (t_3 :: t_1 \rightarrow t_3 \sqsubset t_2)$ ²⁶

In the sequel, we show a partial formalization of this case. We start with the structure that is invariant with respect to the possible types of conjugal relationships:

```

categorizes(ConjugalRelationshipType, ConjugalRelationship)
RelatorKind(ConjugalRelationship)
categorizes(SpouseType, Spouse)
ObjectKind(Person)
Role(Spouse)
Spouse  $\sqsubset$  Person
 $\forall x (x :: \text{Spouse} \rightarrow \exists y (y :: \text{ConjugalRelationship} \wedge \text{mediates}(y, x)))$ 
 $\forall x (x :: \text{ConjugalRelationship} \rightarrow \exists y, z (y :: \text{Spouse} \wedge z :: \text{Spouse} \wedge y \neq z \wedge \text{mediates}(x, y) \wedge \text{mediates}(x, z)))$ 

```

We also need to ensure that `ConjugalRelationships` (at the individual level) respect the invariants specified for `ConjugalRelationshipTypes` and `SpouseTypes`. Thus, we add a constraint that all `ConjugalRelationships` (instances of instances of `ConjugalRelationshipType`) mediate relations that instantiate `SpouseType` that are compatible with their specific `ConjugalRelationshipTypes`.²⁷

$$\forall x, y ((x :: y \wedge y :: \text{ConjugalRelationshipType}) \rightarrow \forall z (\text{mediates}(y, z) \rightarrow \exists t (t :: \text{SpouseType} \wedge \text{compatibleWith}(t, y) \wedge z :: t)))$$

And then introduce two types of `ConjugalRelationships`, in conformance with the formalization so far. First, we introduce `MonogamousHeterosexualMarriage`:

```

MonogamousHeterosexualMarriage :: ConjugalRelationshipType
SubKind(MonogamousHeterosexualMarriage)
MonogamousHeterosexualMarriage  $\sqsubset$  ConjugalRelationship

```

²⁵We refer the reader to Carvalho et al. (2017) which shows the combination of UFO with the Multi-Level Theory (MLT).

²⁶In OntoUML, the «instantiation» stereotype represents both MLT relations of *categorizes* and *powertypeOf*, the later being outside the scope of this example.

²⁷This constraint spans more than two levels of classification and corresponds to what can be obtained with a “regularity attribute” of types (Guizzardi et al., 2015a).

Husband :: SpouseType

Role(Husband)

Husband \sqsubset Spouse

$\forall x(x :: \text{Husband} \rightarrow \exists!y(y :: \text{MonogamousHeterosexualMarriage} \wedge \text{mediates}(y, x))$

Wife :: SpouseType

Role(Wife)

Wife \sqsubset Spouse

$\forall x(x :: \text{Wife} \rightarrow \exists!y(y :: \text{MonogamousHeterosexualMarriage} \wedge \text{mediates}(y, x))$

$\forall x(x :: \text{MonogamousHeterosexualMarriage} \rightarrow \exists!y, !z(y :: \text{Husband} \wedge z :: \text{Wife} \wedge \text{mediates}(x, y) \wedge \text{mediates}(x, z)))$

And, second, MonogamousSamesexMarriage:

MonogamousSamesexMarriage :: ConjugalRelationshipType

SubKind(MonogamousSamesexMarriage)

MonogamousSamesexMarriage \sqsubset ConjugalRelationship

Partner :: SpouseType

Role(Partner)

Partner \sqsubset Spouse

$\forall x(x :: \text{Partner} \rightarrow \exists!y(y :: \text{MonogamousSamesexMarriage} \wedge \text{mediates}(y, x))$

$\forall x(x :: \text{MonogamousSamesexMarriage} \rightarrow \exists!y, !z(y :: \text{Partner} \wedge z :: \text{Partner} \wedge y \neq z \wedge \text{mediates}(x, y) \wedge \text{mediates}(x, z) \wedge ((y :: \text{Man} \wedge z :: \text{Man}) \vee (y :: \text{Woman} \wedge z :: \text{Woman})))$

In the sequence, we present a simulation of such a scenario where in the past world the only instance of *ConjugalRelationshipType* present is the *MonogamicHeterosexualMarriage*, which is also foreseen in the model. In the subsequent world another *ConjugalRelationshipType* is also present, one that has only one compatible *SpouseType* (which can be interpreted as a sort of *Partner* type), and whose instances relate two instances of *Spouse* of the same sex. This simulation shows that *MonogamicSamesexMarriage* is also a possible instance for the model in Figure 15.

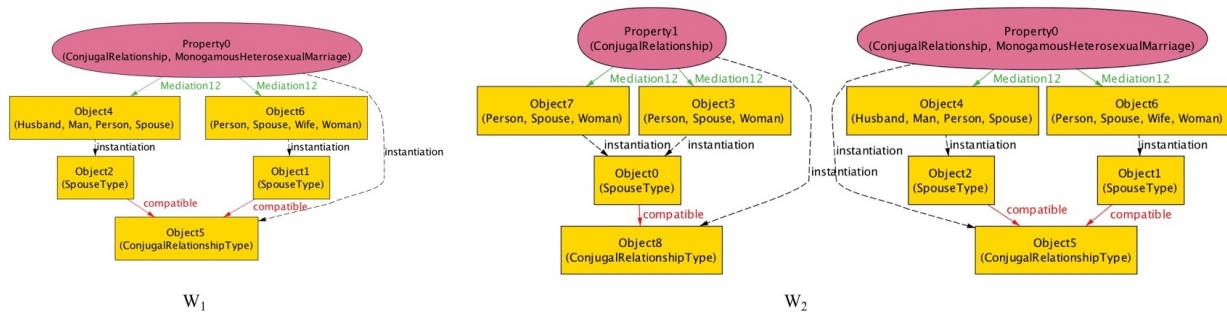


Figure 15. Simulation of evolving notion of *ConjugalRelationship*. In w_1 , the only *ConjugalRelationshipType* present is the one whose instances include *MonogamicHeterosexualMarriage*, e.g., *Property0* (*Object5 :: ConjugalRelationshipType*, *Property0 :: MonogamicHeterosexualMarriage*, *instantiation(Property0, Object5)*). In a subsequent world, it is present a different instance of *ConjugalRelationshipType* — *Object8* — whose instances include same sex conjugal relationships, such as *Property1* (*Object8 :: ConjugalRelationshipType*, *Property1 :: ConjugalRelationship*, *instantiation(Property1, Object8)*). The compatible *SpouseType* to *Object8* — *Object0* — is also different from the ones classifying instances of *Husband* and *Wife* — *Object2* and *Object1* —, what shows that the model also accommodates types such as *Partner* as instances (*Object0 :: SpouseType*, *Object1 :: SpouseType*, *Object2 :: SpouseType*, *Object7 :: Spouse*, *Object3 :: Spouse*, *Object4 :: Husband*, *Object6 :: Wife*, *instantiation(Object7, Object0)*, *instantiation(Object3, Object0)*, *instantiation(Object4, Object2)*, *instantiation(Object6, Object1)*).

We should observe that we are dealing here with a case of *anticipated evolution* (Bennett and Rajlich, 2000), i.e., when it is possible at specification time to foresee that types are likely to change and hence introduce invariant structures that can accommodate the envisioned space of change. *Unanticipated evolution* would effectively require changing the model, which could be obtained by application of model refactoring operations (Sunyé et al., 2001). For example, in the case of the “marriage” scenario, if we had an initial model consisting of a “marriage” relator between a “man” and a “woman,” we could rename it to “heterosexual marriage”, add “same-sex marriage” and introduce “conjugal relationship” as a super type of both of them along with declaring “monogamous heterosexual marriage” and “monogamous same-sex marriage” as instances of “conjugal relationship type” (to foresee other possible changes).

5. Ontology usage and community impact

Over the years, UFO has been used for the development of core and domain ontologies on a wide range of domains, in academic, governmental and industrial contexts Guizzardi et al. (2015b). For instance, it has been used to provide conceptual clarification in domains ranging from natural sciences (e.g., agriculture, bioinformatics, geology) to “purely informational” domains (e.g., telecommunications, multimedia, game design, programming languages), from methodological domains (e.g., design science research) to addressing practical environmental management problems (e.g., simulation for land covering and use, waste management, emergency and disaster management). Moreover, UFO and ontologies built with it have been used to analyze, reengineer, or integrate many modeling languages and standards in different domains (e.g., ArchiMate, ARIS, DEMO, ISO/IEC 24744, ITU-T G.805, BPMN, TOGAF, Tropos and i^* , and UML). In particular, we call attention to a recent industry standard in the tourism area that is entirely based on UFO-B (AlpineBits Alliance, 2021).

Clearly one of the most influential applications of UFO has been on the design of the conceptual modeling language OntoUML and its ecosystem of methodological and computational tools. A recent study shows that UFO is the second-most used foundational ontology in conceptual modeling and the one with the fastest adoption rate (Verdonck and Gailly, 2016). That study also shows that OntoUML is among the most used languages in ontology-driven conceptual modeling (together with UML, (E)ER, OWL, and BPMN). Moreover, empirical evidence shows that OntoUML significantly contributes to improving the quality of conceptual models without requiring an additional effort to producing them. For instance, Verdonck et al. (2019) report on a modeling experiment conducted with 100 participants in two countries showing the advantages (in these respects) of OntoUML when compared to a classical conceptual modeling language (EER – Extended ER).

Currently, the development of UFO-based models through OntoUML is supported by a microservice-based infrastructure. The *OntoUML as a Service* infrastructure (Fonseca et al., 2021b), or OaaS, is designed to decouple model services developed by OntoUML researchers (e.g., transformations, verifications, simulations, verbalizations) and the modeling tools they support. As a result, these *model intelligence services* can be developed independently, making use of programming languages and libraries that best fit the researcher’s needs and preferences, and later integrated into modeling tools, such as UML CASE tools. This is enabled by OaaS’s low requirements on services and tool developers that consists of support to HTTP and JSON used for service request and model serialization respectively. The current implementation of OaaS consists of a JSON Schema specification to govern the JSON serialization of OntoUML models²⁸, a TypeScript library to support the manipulation and serialization of OntoUML models²⁹, an HTTP server that provides multiple model intelligence services³⁰, and a plugin for the UML CASE Visual Paradigm³¹ that extends it with OntoUML modeling capabilities³².

²⁸The `ontouml-schema` project is available at <https://purl.org/krdb-core/ontouml-schema>.

²⁹The `ontouml-js` project is available at <https://purl.org/krdb-core/ontouml-js>.

³⁰The `ontouml-server` is available at <https://purl.org/krdb-core/ontouml-server>.

³¹<https://www.visual-paradigm.com/>

³²The `ontouml-vp-plugin` is available at <https://purl.org/krdb-core/ontouml-plugin>.

Besides OntoUML, UFO has been successfully used in the design of a number of ontologies in a multitude of sub-domains in Software Engineering (Duarte et al., 2018; Henderson-Sellers, 2012; Henderson-Sellers et al., 2014; Kirk and MacDonell, 2016; Maretto and Barcellos, 2014; Morales-Ramirez et al., 2015; Ruy et al., 2016; Shekhovtsov and Mayr, 2014; Sydorov et al., 2019; Bernabé et al., 2019; Duarte et al., 2018, 2021; Guizzardi et al., 2014; Negri et al., 2017). In fact, a number of these ontologies forming SEON (Software Engineering Ontology Network)³³ has been used to address problems such as application integration, semantic annotation of requirements, software quality assurance, code interoperability, among others.

Finally, another area in which UFO had a noticeable impact is domains dealing with legal, social and economic aspects. These include financial accounting (Blums and Weigand, 2021; Fischer-Pauzenberger and Schwaiger, 2017; Fraller, 2019; Laurier et al., 2018) as well as domains dealing with legal relations and contracts (Sharifi et al., 2020; El Ghosh and Abdulrab, 2020, 2021; Griffo et al., 2015; Mário de Oliveira Rodrigues et al., 2020), as well as microeconomic subdomains such as resources and capabilities (Azevedo et al., 2015), competition (Sales et al., 2018b), economic exchanges (Porello et al., 2020), services (Nardi et al., 2015), money (Amaral et al., 2020), trust (Amaral et al., 2019), risk and value (Guarino et al., 2016; Sales et al., 2018a), among others. The perceived usefulness of the ontology in these domains is unsurprising given that most of these domains require a sophisticated modeling of relational aspects. The proper modeling of all the aforementioned phenomena real requires the support of fuller theory of relations, which is one of the distinctive theoretical aspects of UFO.

Acknowledgments

We are grateful to Gerd Wagner, Nicola Guarino, and the members of the Ontology & Conceptual Modeling Research Group (NEMO)—especially, Renata Guizzardi and Ricardo Falbo (*in memoriam*)—for all the years of collaboration in the scope of UFO.

Claudenir M. Fonseca and Giancarlo Guizzardi are supported by the NeXON Project (Free University of Bozen-Bolzano). João Paulo A. Almeida is funded by the Brazilian Research Funding Agencies CNPq (grants number 313687/2020-0 and 407235/2017-5), CAPES (23038.028816/2016-41) Finance Code 001 and FAPES (281/2021). Daniele Porello thanks the European project OntoCommons (GA 958371) for co-funding the writing of this paper.

References

- Albuquerque, A. and Guizzardi, G. (2013). An ontological foundation for conceptual modeling datatypes based on semantic reference spaces. In *IEEE 7th International Conference on Research Challenges in Information Science (RCIS)*, pages 1–12.
- Almeida, J. P. A., Falbo, R. A., and Guizzardi, G. (2019). Events as entities in ontology-driven conceptual modeling. In *International Conference on Conceptual Modeling*, pages 469–483. Springer.
- AlpineBits Alliance (2021). AlpineBits DestinationData 2021-04.
- Amaral, G., Sales, T. P., Guizzardi, G., and Porello, D. (2019). Towards a reference ontology of trust. In Panetto, H. et al., editors, *27th International Conference on Cooperative Information Systems (CoopIS)*, volume 11877 of *Lecture Notes in Computer Science*, pages 3–21, Cham. Springer.
- Amaral, G., Sales, T. P., Guizzardi, G., and Porello, D. (2020). A reference ontology of money and virtual currencies. In *IFIP Working Conference on The Practice of Enterprise Modeling*, pages 228–243. Springer.
- Azevedo, C. L. B., Iacob, M., Almeida, J. P. A., van Sinderen, M., Pires, L. F., and Guizzardi, G. (2015). Modeling resources and capabilities in enterprise architecture: A well-founded ontology-based proposal for ArchiMate. *Information Systems*, 54:235–262.
- Baker, L. R. (2007). *The metaphysics of everyday life*. Cambridge: Cambridge University Press.
- Benevides, A. B., Bourguet, J.-R., Guizzardi, G., Penaloza, R., and Almeida, J. P. A. (2019). Representing a reference foundational ontology of events in SROIQ. *Applied Ontology*, 14(3):293–334.
- Benevides, A. B., Guizzardi, G., Braga, B. F. B., and Almeida, J. P. A. (2010). Validating modal aspects of OntoUML conceptual models using automatically generated visual world structures. *Journal of Universal Computer Science*, 16(20):2904–2933.

³³<https://dev.nemo.inf.ufes.br/seon/>

- Bennett, K. H. and Rajlich, V. T. (2000). Software maintenance and evolution: A roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, pages 73–87. ACM.
- Bera, P. and Wand, Y. (2004). Analyzing OWL using a philosophy-based ontology. In Varzi, A. and Vieu, L., editors, *3rd International Conference on Formal Ontology in Information Systems (FOIS)*, pages 353–62. IOS Press.
- Bernabé, C. H., Silva Souza, V. E., Almeida Falbo, R. d., Guizzardi, R. S. S., and Silva, C. (2019). GORO 2.0: Evolving an ontology for goal-oriented requirements engineering. In Guizzardi, G., Gailly, F., and Suzana Pitangueira Maciel, R., editors, *Advances in Conceptual Modeling. ER 2019*, volume 11787 of *Lecture Notes in Computer Science*, pages 169–179. Springer.
- Blums, I. and Weigand, H. (2017). Financial reporting by a shared ledger. In *8th International Workshop on Formal Ontologies Meet Industry (FOMI)*.
- Blums, I. and Weigand, H. (2021). Conceptualizing social relators and economic exchange contracts for reporting purposes. In *International Workshop on Value Modelling and Business Ontologies (VMBO)*, volume 2835.
- Bradley, F. (1893). *Appearance and reality: a metaphysical essay*. Swan Sonnenschein & Co.
- Bunge, M. (1977). *Treatise on Basic Philosophy. Ontology I: The Furniture of the World.*, volume 3. Reidel Publishing Company.
- Carmo, A. P., Zamperini, T., Mello, M. R. S., Leal, A. L. C., and Garcia, A. S. (2017). Ontologia das coisas para espaços inteligentes baseados em visão computacional. In *9th Seminar of Ontology Research in Brazil (ONTOBRAS)*.
- Carvalho, V. A. and Almeida, J. P. A. (2018). Toward a well-founded theory for multi-level conceptual modeling. *Software & Systems Modeling*, 17(1):205–231.
- Carvalho, V. A., Almeida, J. P. A., Fonseca, C. M., and Guizzardi, G. (2017). Multi-level ontology-based conceptual modeling. *Data & Knowledge Engineering*, 109:3–24.
- Davidson, D. (2001). *Essays on actions and events: Philosophical essays*, volume 1. Oxford University Press on Demand.
- de Oliveira Bringuente, A. C., de Almeida Falbo, R., and Guizzardi, G. (2011). Using a foundational ontology for reengineering a software process ontology. *Journal of Information and Data Management*, 2(3):511–511.
- Duarte, B. B., de Almeida Falbo, R., Guizzardi, G., Guizzardi, R., and Souza, V. E. S. (2021). An ontological analysis of software system anomalies and their associated risks. *Data & Knowledge Engineering*, 134:101892.
- Duarte, B. B., Leal, A. L. C., Falbo, R. A., Guizzardi, G., Guizzardi, R. S. S., and Souza, V. E. S. (2018). Ontological foundations for software requirements with a focus on requirements at runtime. *Applied Ontology*, 13(2):73–105.
- El Ghosh, M. and Abdulrab, H. (2020). Ontology-based liability decision support in the international maritime law. In *Legal Knowledge and Information Systems*, pages 273–276. IOS Press.
- El Ghosh, M. and Abdulrab, H. (2021). Cargo-s: A pattern-based well-founded legal domain ontology for the traceability of goods in logistic sea corridors. *Applied Ontology*, pages 1–40.
- Evermann, J. and Wand, Y. (2001). Towards ontologically based semantics for UML constructs. In Kunii, H. S., Jajodia, S., and Sølvberg, A., editors, *Conceptual Modeling. ER 2001*, volume 2224 of *Lecture Notes in Computer Science*, pages 354–367. Springer.
- Fischer-Pauzenberger, C. and Schwaiger, W. S. (2017). The ontorea accounting model: Ontology-based modeling of the accounting domain. *Complex Systems Informatics and Modeling Quarterly*, 11:20–37.
- Fitting, M. and Mendelsohn, R. L. (2012). *First-order modal logic*, volume 277. Springer Science & Business Media.
- Fonseca, C. M., Almeida, J. P. A., Guizzardi, G., and Carvalho, V. A. (2021a). Multi-level conceptual modeling: Theory, language and application. *Data & Knowledge Engineering*, 134:101894.
- Fonseca, C. M., Porello, D., Guizzardi, G., Almeida, J. P. A., and Guarino, N. (2019). Relations in ontology-driven conceptual modeling. In *International Conference on Conceptual Modeling*, pages 28–42. Springer.
- Fonseca, C. M., Sales, T. P., Viola, V., Fonseca, L. B. R., Guizzardi, G., and Almeida, J. P. A. (2021b). Ontology-driven conceptual modeling as a service. In *11th International Workshop on Formal Ontologies meet Industry (FOMI 2021)*.
- Fraller, C. (2019). *Flexible budgeting in an activity-based costing framework: from conceptual modeling to prototypical implementation*. PhD thesis, Wien.
- Gärdenfors, P. (2004). *Conceptual spaces: The geometry of thought*. MIT press.
- Griffo, C., Almeida, J. P. A., and Guizzardi, G. (2015). Towards a legal core ontology based on Alexy's theory of fundamental rights. In *Multilingual Workshop on Artificial Intelligence and Law*.
- Guarino, N., Andersson, B., Johannesson, P., and Livieri, B. (2016). Towards an ontology of value ascription. In *9th International Conference on Formal Ontology in Information Systems (FOIS)*, volume 283, page 331. IOS Press.
- Guarino, N. and Guizzardi, G. (2015). "We need to discuss the relationship": Revisiting relationships as modeling constructs. In *Advanced Information Systems Engineering. CAiSE 2015*, volume 9097 of *Lecture Notes in Computer Science*, pages 279–294.
- Guarino, N. and Guizzardi, G. (2016). Relationships and events: Towards a general theory of reification and truthmaking. In Adorni, G., Cagnoni, S., Gori, M., and Maratea, M., editors, *Advances in Artificial Intelligence. AI*IA 2016*, volume 10037 of *Lecture Notes in Computer Science*, pages 237–249.
- Guarino, N., Guizzardi, G., and Mylopoulos, J. (2020). On the philosophical foundations of conceptual models. *Information Modelling and Knowledge Bases*, 31(321):1.
- Guarino, N., Porello, D., and Guizzardi, G. (2019). On weak truthmaking. In Barton, A., Seppälä, S., and Porello, D., editors, *Proceedings of the Joint Ontology Workshops 2019 Episode V: The Styrian Autumn of Ontology, Graz, Austria, September 23-25, 2019*, volume 2518 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Guarino, N. and Welty, C. A. (2009). An overview of OntoClean. In *Handbook on Ontologies*, International Handbooks on Information Systems, pages 201–220. Springer.
- Guerson, J., Sales, T. P., Guizzardi, G., and Almeida, J. P. A. (2015). OntoUML Lightweight Editor: a model-based environment to build, evaluate and implement reference ontologies. In *19th IEEE International Enterprise Distributed Object Computing Workshops*, pages 144–147. IEEE.
- Guizzardi, G. (2005). *Ontological foundations for structural conceptual models*. CTIT, Centre for Telematics and Information Technology, Enschede.

- Guizzardi, G., Almeida, J. P. A., Guarino, N., and Carvalho, V. A. (2015a). Towards an ontological analysis of powertypes. In Papini, O. et al., editors, *Joint Ontology Workshops (JOWO)*, volume 1517. CEUR-WS.org.
- Guizzardi, G., Fonseca, C. M., Almeida, J. P. A., Sales, T. P., Benevides, A. B., and Porello, D. (2021). Types and taxonomic structures in conceptual modeling: A novel ontological theory and engineering support. *Data & Knowledge Engineering*, 134.
- Guizzardi, G., Guarino, N., and Almeida, J. P. A. (2016). Ontological considerations about the representation of events and endurants in business models. In Rosa, M. L., Loos, P., and Pastor, O., editors, *Business Process Management. BPM 2016*, volume 9850 of *Lecture Notes in Computer Science*, pages 20–36. Springer.
- Guizzardi, G., Masolo, C., and Borgo, S. (2006). In defense of a trope-based ontology for conceptual modeling: An example with the foundations of attributes, weak entities and datatypes. In Persson, A. and Stirna, J., editors, *Conceptual Modeling. ER 2006*, volume 3084 of *Lecture Notes in Computer Science*, pages 112–125. Springer.
- Guizzardi, G. and Wagner, G. (2004). A Unified Foundational Ontology and some applications of it in business modeling. In Missikoff, M., editor, *Enterprise Modelling and Ontologies for Interoperability. EMOI - INTEROP 2004*, volume 125. CEUR-WS.org.
- Guizzardi, G., Wagner, G., Almeida, J. P. A., and Guizzardi, R. S. S. (2015b). Towards ontological foundations for conceptual modeling: The Unified Foundational Ontology (UFO) story. *Applied Ontology*, 10(3-4):259–271.
- Guizzardi, G., Wagner, G., Falbo, R. A., Guizzardi, R. S. S., and Almeida, J. P. A. (2013). Towards ontological foundations for the conceptual modeling of events. In Ng, W., Storey, V. C., and Trujillo, J. C., editors, *Conceptual Modeling. ER 2013*, volume 8217 of *Lecture Notes in Computer Science*, pages 327–341. Springer.
- Guizzardi, G. and Zamborlini, V. (2014). Using a trope-based foundational ontology for bridging different areas of concern in ontology-driven conceptual modeling. *Science of Computer Programming*, 96:417–443.
- Guizzardi, R. S. S., Li, F., Borgida, A., Guizzardi, G., Horkoff, J., and Mylopoulos, J. (2014). An ontological interpretation of non-functional requirements. In *8th International Conference on Formal Ontology in Information Systems (FOIS)*, pages 344–357.
- Heller, B. and Herre, H. (2004). Ontological categories in gol. *Axiomathes*, 14(1):57–76.
- Henderson-Sellers, B. (2012). *On the Mathematics of Modelling, Metamodeling, Ontologies and Modelling Languages*. Springer Briefs in Computer Science. Springer.
- Henderson-Sellers, B., Gonzalez-Perez, C., McBride, T., and Low, G. (2014). An ontology for ISO software engineering standards: 1) creating the infrastructure. *Computer Standards & Interfaces*, 36(3):563–576.
- Herre, H., Heller, B., Burek, P., Loebe, F., and Michalek, H. (2004). General ontological language: A formal framework for building and representing ontologies (version 1.0). *Onto-Med Report 7*, Reseach Group Ontologies in Medicine, University of Leipzig.
- Hitchman, S. (2003). An interpretive study of how practitioners use entity-relationship modelling in a ternary relationship situation. *Communications of the Association for Information Systems*, 11:26.
- Hovda, P. (2009). What is classical mereology? *Journal of Philosophical Logic*, 38(1):55–82.
- Keil, F. C. (1992). *Concepts, kinds, and cognitive development*. mit Press.
- Kirk, D. and MacDonell, S. (2016). An ontological analysis of a proposed theory for software development. In Lorenz, P., Cardoso, J., Maciaszek, L. A., and van Sinderen, M., editors, *Software Technologies. ICSoft 2015*, volume 586 of *Communications in Computer and Information Science*, pages 155–171. Cham. Springer.
- Laurier, W., Kiehn, J., and Polovina, S. (2018). Rea 2: A unified formalisation of the resource-event-agent ontology. *Applied Ontology*, 13(3):201–224.
- Lowe, E. J. and Lowe, E. J. (2006). *The four-category ontology: A metaphysical foundation for natural science*. Oxford University Press.
- Maretto, C. X. and Barcellos, M. P. (2014). A levels-based approach for defining software measurement architectures. *CLEI Electronic Journal*, 17(3).
- Mário de Oliveira Rodrigues, C., Bezerra, C., Freitas, F., and Oliveira, I. (2020). Handling crimes of omission by reconciling a criminal core ontology with UFO. *Applied Ontology*, 15(1):7–39.
- Masolo, C., Borgo, S., Gangemi, A., Guarino, N., and Oltramari, A. (2003). WonderWeb Deliverable D18. Technical report, CNR.
- Moltmann, F. (2020). Variable objects and truthmaking. In *Metaphysics, Meaning, and Modality*, pages 368–394. Oxford University Press.
- Morales-Ramirez, I., Perini, A., and Guizzardi, R. S. (2015). An ontology of online user feedback in software engineering. *Applied Ontology*, 10(3-4):297–330.
- Nardi, J. C., Falbo, R. A., Almeida, J. P. A., Guizzardi, G., Pires, L. F., van Sinderen, M., Guarino, N., and Fonseca, C. M. (2015). A commitment-based reference ontology for services. *Information Systems*, 54:263–288.
- Negri, P. P., Souza, V. E. S., Leal, A. L. C., Falbo, R. A., and Guizzardi, G. (2017). Towards an ontology of goal-oriented requirements. In *20th Iberoamerican Conference on Software Engineering (CIBSE), Buenos Aires, Argentina, May 22-23, 2017.*, pages 469–482.
- Parsons, T. (1990). *Events in the Semantics of English*, volume 5. Cambridge, Ma: MIT Press.
- Porello, D., Guizzardi, G., Sales, T. P., and Amaral, G. (2020). A core ontology for economic exchanges. In Dobbie, G., Frank, U., Kappel, G., Liddle, S. W., and Mayr, H. C., editors, *Conceptual Modeling*, volume 12400 of *Lecture Notes in Computer Science*, pages 364–374. Springer.
- Quine, W. V. O. (1937). New foundations for mathematical logic. *The American mathematical monthly*, 44(2):70–80.
- Ruy, F. B., Falbo, R. A., Barcellos, M. P., Costa, S. D., and Guizzardi, G. (2016). SEON: A software engineering ontology network. In *20th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, pages 527–542.

- Sales, T. P., Baião, F., Guizzardi, G., Guarino, N., and Mylopoulos, J. (2018a). The common ontology of value and risk. In Trujillo, J. C. et al., editors, *Conceptual Modeling. ER 2018*, volume 11157 of *Lecture Notes in Computer Science*, pages 121–135. Cham. Springer.
- Sales, T. P. and Guizzardi, G. (2015). Ontological anti-patterns: empirically uncovered error-prone structures in ontology-driven conceptual models. *Data & Knowledge Engineering*, 99:72–104.
- Sales, T. P., Porello, D., Guarino, N., Guizzardi, G., and Mylopoulos, J. (2018b). Ontological foundations of competition. In Borgo, S., Hitzler, P., and Kutz, O., editors, *10th International Conference on Formal Ontology in Information Systems (FOIS)*, volume 306 of *Frontiers in Artificial Intelligence and Applications*, pages 96–109. IOS Press.
- Sharifi, S., Parvizimosaed, A., Amyot, D., Logrippo, L., and Mylopoulos, J. (2020). Symboleo: Towards a specification language for legal contracts. In *2020 IEEE 28th International Requirements Engineering Conference (RE)*, pages 364–369. IEEE.
- Shekhovtsov, V. A. and Mayr, H. C. (2014). Managing quality related information in software development processes. In *CAiSE Forum and Doctoral Consortium 2014*, volume 1164, pages 73–80. CEUR-WS.org.
- Smith, B. C. (2019). *The promise of artificial intelligence: reckoning and judgment*. Mit Press.
- Stout, R. (1997). Processes. *Philosophy*, 72(279):19–27.
- Strawson, P. F. (1959). An essay in descriptive metaphysics, chapter individuals.
- Sunyé, G., Pollet, D., Le Traon, Y., and Jézéquel, J.-M. (2001). Refactoring UML models. In Gogolla, M. and Kobryn, C., editors, *UML 2001 – The Unified Modeling Language. Modeling Languages, Concepts, and Tools*, volume 2185 of *Lecture Notes in Computer Science*, pages 134–148. Springer.
- Sutcliffe, G. (2017). The TPTP Problem Library and Associated Infrastructure. From CNF to TH0, TPTP v6.4.0. *Journal of Automated Reasoning*, 59(4):483–502.
- Sydorov, N., Sydorova, N., Sydorov, E., Cholyskhina, O., and Batsurovska, I. (2019). Development of an approach to using a style in software engineering. *Eastern-European Journal of Enterprise Technologies*, 4(2):41–51.
- Thomson, J. J. (1998). The statue and the clay. *Noûs*, 32(2):149–173.
- Verdonck, M. and Gailly, F. (2016). Insights on the use and application of ontology and conceptual modeling languages in ontology-driven conceptual modeling. In Comyn-Wattiau, I. et al., editors, *Conceptual Modeling. ER 2016*, volume 9974 of *Lecture Notes in Computer Science*. Springer.
- Verdonck, M., Gailly, F., Pergl, R., Guizzardi, G., Martins, B., and Pastor, O. (2019). Comparing traditional conceptual modeling with ontology-driven conceptual modeling: an empirical study. *Information Systems*, 81:92–103.
- Veres, C. and Hitchman, S. (2002). Using psychology to understand conceptual modelling. In *ECIS*, pages 473–481.
- Veres, C. and Mansson, G. (2005). Cognition and modeling: Foundations for research and practice. *Journal of Information Technology Theory and Application*, 7(1):9.
- Wand, Y., Storey, V. C., and Weber, R. (1999). An ontological analysis of the relationship construct in conceptual modeling. *ACM Transactions on Database Systems*, 24(4):494–528.
- Wand, Y. and Weber, R. (1989). An ontological evaluation of systems analysis and design methods. In Falkenberg, E. and Lindgreen, P., editors, *Information Systems Concepts – An In-Depth Analysis*, pages 79–108. North-Holland.
- Wand, Y. and Weber, R. (1990). An ontological model of an information system. *IEEE Transactions on Software Engineering*, 16(11):1282–1292.
- Weber, R. (1997). *Ontological foundations of information systems*. Coopers & Lybrand and the Accounting Association of Australia and New Zealand Melbourne.
- Weber, R. and Zhang, Y. (1991). An ontological evaluation of niam’s grammar for conceptual schema diagrams. In *ICIS*, pages 75–82. Association for Information Systems.
- Zamborlini, V., Betti, A., and van den Heuvel, C. (2017). Toward a core conceptual model for (im)material cultural heritage in the golden agents project. In *1st International Workshop on Understanding Events Semantics in Cultural Heritage*.