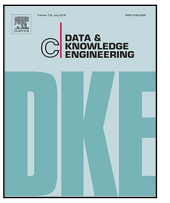
Data & Knowledge Engineering 134 (2021) 101891

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak



Types and taxonomic structures in conceptual modeling: A novel ontological theory and engineering support

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

[a] Free University of Bozen-Bolzano, Italy
[b] University of Twente, The Netherlands
[c] Federal University of Espírito Santo, Brazil
[d] Italian National Research Council (CNR), Italy

ARTICLE INFO

Keywords:
Taxonomic structures
Ontology-driven conceptual modeling
OntoUML
Unified foundational ontology (UFO)
Conceptual modeling
Ontologies

ABSTRACT

Types are fundamental for conceptual modeling and knowledge representation, being an essential construct in all major modeling languages in these fields. Despite that, from an ontological and cognitive point of view, there has been a lack of theoretical support for precisely defining a consensual view on types. As a consequence, there has been a lack of precise methodological support for users when choosing the best way to model general terms representing types that appear in a domain, and for building sound taxonomic structures involving them. For over a decade now, a community of researchers has contributed to the development of the Unified Foundational Ontology (UFO) - aimed at providing foundations for all major conceptual modeling constructs. At the core of this enterprise, there has been a theory of types specially designed to address these issues. This theory is ontologically well-founded, psychologically informed, and formally characterized. These results have led to the development of a Conceptual Modelling language dubbed OntoUML, reflecting the ontological micro-theories comprising UFO. Over the years, UFO and OntoUML have been successfully employed on conceptual model design in a variety of domains including academic, industrial, and governmental settings. These experiences exposed improvement opportunities for both the OntoUML language and its underlying theory, UFO. In this paper, we revise the theory of types in UFO in response to empirical evidence. The new version of this theory shows that many of OntoUML's meta-types (e.g. kind, role, phase, mixin) should be considered not as restricted to substantial types but instead should be applied to model endurant types in general, including relator types, quality types, and mode types. We also contribute with a formal characterization of this fragment of the theory, which is then used to advance a new metamodel for OntoUML (termed OntoUML 2). To demonstrate that the benefits of this approach are extended beyond OntoUML, the proposed formal theory is then employed to support the definition of UFO-based lightweight Semantic Web ontologies with ontological constraint checking in OWL. Additionally, we report on empirical evidence from the literature, mainly from cognitive psychology but also from linguistics, supporting some of the key claims made by this theory. Finally, we propose a computational support for this updated metamodel.

* Corresponding author.
E-mail addresses: giancarlo.guizzardi@unibz.it (G. Guizzardi), cmoraisfonseca@unibz.it (C.M. Fonseca), jpalmeida@ieee.org (J.P.A. Almeida), tiago.princesales@unibz.it (T.P. Sales), alessander.benevides@vilavelha.es.gov.br (A.B. Benevides), daniele.porello@cnr.it (D. Porello).

https://doi.org/10.1016/j.datak.2021.101891
Received 13 February 2021; Accepted 10 May 2021
Available online 13 May 2021

1. Introduction

Types are fundamental for conceptual modeling and knowledge representation, being an essential construct in all major modeling languages in these fields. In general, monadic types used in structural conceptual models stand for universals whose instances are *entities* or *objects*, i.e., entities that exist in time possibly changing their properties while maintaining their identities. In the philosophical literature, these entities are also termed *things* [1], *continuants* [2], or *endurants* [3]. Once more, the intuition behind these terms is that these entities continue to exist or endure through qualitative changes, in contrast with *perdurants* or *occurrences* (e.g., events, processes), which unfold in time accumulating temporal parts.

In the literature and practice of conceptual modeling, a set of primitives has often been used to represent distinctions among different categories of types. For example, we have the notion of *natural type* in [1], the notions of *static types*, *dynamic types*, and *roles* in [4], and the notion of *mixin* in [5]. However, traditionally, there was a lack of methodological support for helping the user of the language to decide how to represent elements that denote general properties in a given subject domain. There was also much debate and lack of consensus on the meaning of these categories [4,6–8]. As a consequence, modeling choices were often made in an *ad hoc* manner, and model interoperability was hindered. Furthermore, types typically appear in conceptual models forming taxonomic structures via *subtyping* relations and *generalization sets* constraining relations among relations. Once again, historically, there was a lack of theoretical support for building sound taxonomic structures involving all these different categories of types.

A pioneering contribution addressing these problems and proposing a richer set of ontological distinctions among categories of types, was the work of Nicola Guarino in [9], and later in the typology of types underlying the OntoClean methodology [10]. In [11], Guizzardi, Wagner, van Sinderen, and Guarino propose a revision to this typology, extending it into an ontological theory particularly created to provide foundations for conceptual modeling types. Their work aimed at addressing specific recurrent problems in the conceptual modeling literature (i.e., *problem patterns*), and was strongly influenced by research in *descriptive metaphysics* [2], philosophical logics [12] and, in particular, by a series of psychological claims put forth by the cognitive psychologist John Macnamara in [13]. In addition to the ontology work, in that paper, they systematically employ their theory to evaluate and redesign a fragment of UML class diagrams, formally differentiating these categories of types, and providing formal rules for creating sound taxonomic structures involving them. Together with [14], this work represented the beginning of the Unified Foundational Ontology (UFO) [3,15] – a foundational ontology specially built to provide foundations underlying all major conceptual modeling constructs, as well as of the UFO-based ontology-driven conceptual modeling (ODCM) language OntoUML [3,15].

For more than a decade now, UFO and OntoUML have been successfully employed in academic, industrial, and governmental settings to create conceptual models in plenty of different domains, including Geology, Biodiversity Management, Organ Donation, Petroleum Reservoir Modeling, Disaster Management, Context Modeling, Datawarehousing, Enterprise Architecture, Data Provenance, Measurement, Logistics, Complex Media Management, Telecommunications, Heart Electrophysiology, among many others [15]. In fact, research shows that they are among the most used foundational ontology and modeling language in the ODCM literature, respectively [16]. 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, the work of [17] reports 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).

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 foundations. In particular, we have managed to observe numerous ways in which people would slightly subvert the syntax of the language, ultimately creating what we could call *"systematic subversions"* of the language [15]. These "subversions" would (purposefully) produce models that were syntactically incorrect, but which were needed to express the intended characterization of their underlying conceptualizations that could not be expressed otherwise. Moreover, they were "systematic" because they would recur in the works of different authors that would, independently of each other, subvert the language in the same manner and with the same modeling intention. One of these "language subversions" addresses the exact topic of this paper, the theory of types and taxonomic structures of UFO, and hence, the modeling support for these notions in OntoUML.

As we previously mentioned, structural conceptual modeling languages are designed to model types whose instances are endurants. However, in the original version of UFO's theory of types, many of these meta-types present in OntoUML (e.g., kinds, role, phases, mixins) were restricted to the modeling of *substantial types*, i.e., types whose instances are independent endurants. Intuitively, what in ordinary language we call *objects*. For instance, one could represent that "Person" is the *Kind* of the entity Mick Jagger, but that he is also in a "Senior Citizen" *Phase*, that he plays the *Role* of "Singer" and "Knight of the British Empire" in the scope of certain relations to The Rolling Stones and the Order of the British Empire, respectively. However, one could not model that the relationship between Giovanni and the United Nations (UN) is of the *Kind* "Employment", that it is currently in a "Tenured" *Phase*, and that it can play the *Role* of "Legal Grounds" for his visa application.

Consciously ignoring this restriction, users of the language started to systematically employ these meta-type distinctions to other types of endurants, in particular, to *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 "subversions" were needed to capture subtle aspects of domains such as value, service, and economic exchange (among many others) [18–21]. This called our attention to the fact that, like full-fledged endurants, qualities, modes, and relators are also subject of both essential and accidental properties, and as such, they can also instantiate contingent types, such as phases and roles, and that, in complex domains, their types can also be involved in sophisticated taxonomic structures. In other words, meta-types such as phases, roles, role mixins, mixins, categories, etc. are meta-types of endurants, in general, and not only of substantials.

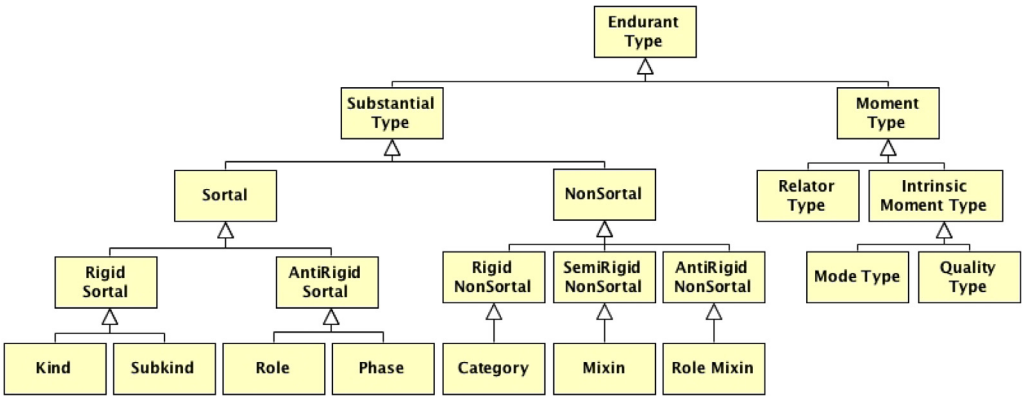


Fig. 1. A taxonomy of endurant types in UFO.

The contributions of this paper are four-fold. First, we propose a new formal theory of endurant types and taxonomic structures for UFO. Although developed in the framework of UFO, this theory amounts to a contribution to ODCM, more broadly. In particular, we can influence approaches such as OntoClean and ORM (Object-Role Modeling), which are sensitive to these matters. Second, following the same ontology-based language engineering approach that was used to create the original version of OntoUML [3], we design the language's (henceforth termed OntoUML 2) new metamodel by employing this new formalization. A UML profile.[1] for class diagrams is then provided, capable of enforcing syntactic constraints that arise from theorems of the formal theory. Third, to demonstrate that the benefits of this approach are extended beyond OntoUML, the proposed formal theory is employed to support the definition of UFO-based lightweight Semantic Web ontologies. SPARQL queries are provided to detect violations of the rules imposed by the formal theory, leveraging the facilities that were only available to OntoUML users to OWL (Web Ontology Language) users as well. Fourth, we present a software tool capable of supporting model verification for OntoUML 2.

This work builds on the work presented in [23], extending it significantly in terms of the support provided for OWL ontologies, as well as tool support. In addition, we present here a formalization in the Object Constraint Language (OCL) [24] for the whole set of constraints governing this profile, which was only discussed informally in the original paper. Finally, by reviewing the proper literature of cognitive psychology, we elaborate on existing empirical support for many of the fundamental choices underlying our theory.

The remainder of this paper is organized as follows: Section 2 presents the background on OntoUML and UFO; Section 3 discusses and formalizes the changes on the underlying UFO theory; Section 4 introduces a new version of OntoUML, presenting its constructs and syntactic constraints; Section 5 introduces the OWL realization of the UFO taxonomy of endurant types and corresponding rules for UFO-based Semantic Web ontologies; Section 6 presents the tool support engineering for conceptual models based on this theory; Section 7 elaborates on cognitive psychology support for fundamental aspects of our theory, and finally, Section 8 concludes the paper with our final considerations.

2. Background: UFO and OntoUML

OntoUML, as all structural conceptual modeling languages is meant to represent type-level structures whose instances are *endurants* (object-like entities), i.e., they are meant to model *Endurant Types* and their type-level relations. Fig. 1 depicts the hierarchy of Endurant Types in UFO.

UFO distinguishes endurant types into substantial types and moment types. Naturally, these are sorts of types whose instances are substantials and moments [3], respectively. Substantials are existentially independent objects such as John Lennon, the Moon, a dog, and a car. Moments, in contrast, are *existentially dependent* individuals, such as:

(a) Sofia's capacity to speak Italian (which depends on her), and
(b) the marriage between John and Yoko (which depends on both John and Yoko).

Moments of type (a) are termed modes; those of type (b) are termed relators. Relators are individuals with the power of connecting entities. For example, an enrollment relator connects an individual playing the Student role with an Educational Institution. Every instance of a relator type is existentially dependent on at least two distinct entities. Moreover, relators are typically composed of modes, for example, in the way that the marriage between John and Mary is composed of their mutual commitments and claims. Furthermore, there is a third sort of moments termed qualities. Qualities are individual moments that can be mapped to some quality space, e.g., an apple's color which may change from green to red while maintaining its identity [3].

[1] For UML profiles, see [22, §12.3].

Concerning the substantial type hierarchy, sortal types are the ones that either provide or carry a uniform *principle of identity* for their instances.[2] A principle of identity regarding a sortal S makes explicit the properties that no two instances of S can have in common, as such properties uniquely identify S instances. In particular, it also informs which changes an individual can undergo without changing its identity, i.e., while remaining the same. Within the category of sortals, we can further distinguish between rigid and anti-rigid. A rigid type is one that classifies its instances necessarily (in the modal sense), i.e., the instances of that type cannot cease to be so without ceasing to exist. Anti-rigidity, in contrast, characterizes a type whose instances can move in and out of its extension without altering their identity [3]. For instance, contrast the rigid type Person with the anti-rigid types Student or Husband. While the same individual John never ceases to be an instance of Person, he can move in and out of the extension of Student or Husband, depending on whether he enrolls in/finishes college or marries/divorces, respectively.

Kinds (also termed *Substance Sortals* [12]) are sortal rigid types that provide a uniform principle of identity for their instances (e.g., Person).[3] Subkinds are sortal rigid types that carry the principle of identity supplied by a unique Kind (e.g., a kind Person can have the subkinds Man and Woman that carry the principle of identity provided by Person). Concerning anti-rigid sortal types, we have the distinction between roles and phases. Phases are relationally independent types defined by contingent but intrinsic instantiation conditions [3]. For example, a Child is a phase of Person, instantiated by instances of persons who have the intrinsic property of being less than 12 years old. Roles, in contrast, are relationally dependent types, capturing relational properties shared by instances of a given kind, i.e., putting it baldly: entities play roles when related to other entities via the so-called material relations (e.g., in the way some plays the role Husband when connected via the material relation of "being married to" with someone playing the role of Wife). Since each individual in the universe of discourse must obey exactly one principle of identity, which, in turn, is provided by a Kind, each sortal hierarchy has a unique Kind at the top, also referred to as *ultimate sortal* [3].

Non-sortals (also called *dispersive types* [3]) are types that aggregate properties that are common to different sortals, i.e., that ultimately classify entities that are of different Kinds. Non-sortals do not provide a uniform principle of identity for their instances and, hence, these types cannot be directly instantiated [2,12,13,26][4]; instead, they just classify things that share common properties but which obey different principles of identity. Furniture is an example of non-sortal that aggregates properties of Table, Chair, and so on. Other examples include Work of Art (including paintings, music compositions, statues), Insurable Item (including works of arts, buildings, cars, body parts) and Legal Entity (including people, organizations, contracts). The meta-properties of rigidity and anti-rigidity can also be applied to distinguish different types of Non-Sortals. A Category represents a rigid and relationally independent non-sortal, i.e., a dispersive type that aggregates essential properties that are common to different rigid sortals [3] (e.g., Physical Object aggregates essential properties of tables, cars, glasses). A Role Mixin represents an anti-rigid and relationally dependent non-sortal, i.e., a dispersive type that aggregates properties that are common to different Roles (e.g., the type Customer that aggregates properties of individual customers and corporate customers) [3]. Although not prescribed in the original set of UFO endurant types, over the years, the notion of a Phase Mixin emerged as a useful notion that was found missing by different authors [27,28]. A *Phase Mixin* represents an anti-rigid and relationally independent non-sortal, i.e., a dispersive type that aggregates properties that are common to different Phases (e.g., the type Active Agent that aggregates properties of living people and active organizations). Finally, a Mixin is a non-sortal that represents properties shared by things of different kinds but which are essential to some of these instances and accidental to some others. For example, the type Insured Item can be essential to cars (suppose all cars must be insured) while being accidental to houses (e.g., houses can be insured but are not necessarily insured).

The leaf ontological distinctions represented in Fig. 1 as well as their corresponding axiomatization are reflected as modeling constructs in OntoUML [3]. An example of a model illustrating these notions is presented in Fig. 2.

As one can observe in Fig. 1, the meta-types discussed in this section specialize *Substantial Type* and, thus, are disjoint of all subcategories of *Moment Types*. In other words, in traditional OntoUML models such as the one of Fig. 2, these meta-types are restricted to modeling types whose instances are substantials (e.g., Organization, Family, Person, Country). As we previously discussed, this limitation appears in our empirical data as a *"systematic subversion"* of the syntactical rules of the language [15]. Once more, as full-fledged endurants, moments can have essential and accidental properties and, as such, they instantiate kinds and subkinds, as well as phases and roles. Once we allow for both sort of modal properties to cross the boundaries of (moment) types (as they do, as we exemplify in Section 4), then the existence of rigid and anti-rigid non-sortals logically follows. As discussed in Section 7, this empirical evidence conforms with other evidence found in the literature. The novel theory of types presented in the next section addresses exactly this issue.

3. A new formal theory of endurant types

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). Types and their instances are connected by instantiation relations (symbolized as "::"). For our purposes, the first-order modal logic QS5 with the Barcan formula and its converse suffices [29]. That means that we assume a *fixed* domain of entities for every possible world, what is traditionally associated with a *possibilistic* view of the entities of the domain, i.e., the domain includes all the *possibilia*. In the following formulas, we drop both the universal quantifier and the

[2] This is one the most supported claims in philosophy of language [2,12,25], and one that finds very strong empirical support in cognitive psychology (see discussion on Section 7). In [13], Macnamara writes: *"We cannot conceptually grasp an individual without the support of a count noun" (psychological claim 2)*. As he defends in the same volume, count nouns are the linguist counterparts of Sortals.

[3] In the philosophical literature, rigidity is also termed *modal invariance* of the type extension [12]. Regarding this, Macnamara writes: *"We employ the word 'dog' to refer to the dogs of the past and to future dogs as well as to present day dogs" (psychological claim 4) [13]*.

[4] *"We cannot conceptually grasp an individual in a universal kind supposedly denoted by the count noun 'thing' or 'object'." (psychological claim 3) [13]*.

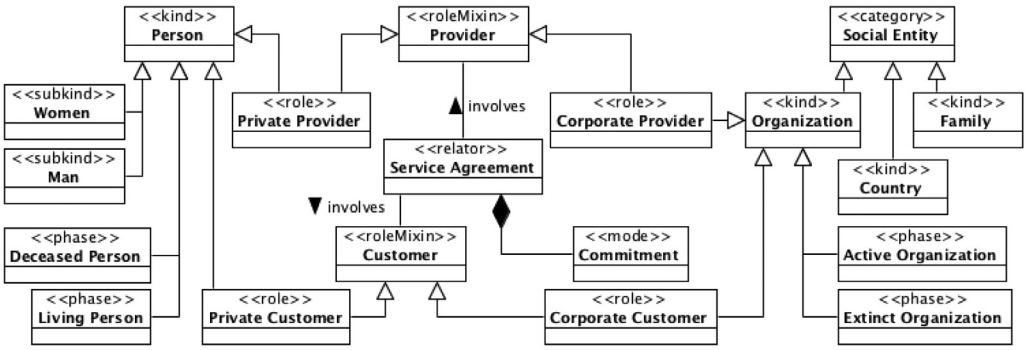


Fig. 2. OntoUML example.

necessity operator in case their scope takes the full formula. In what follows, (a*) and (t*) patterns refer to axioms and theorems, respectively, and the modal operators of *necessity* ("$\Box$") and *possibility* ("$\Diamond$") are used with their usual (QS5) meaning. This theory has been specified and verified in TPTP (Thousands of Problems for Theorem Provers),[5] being automatically proven with provers SPASS 3.9 and Z3 4.4.1.[6]

Firstly, *types* are implicitly defined as those entities that are possibly instantiated (a1), while *individuals* are those necessarily not instantiated (a2). Since we are only concerned with first-order types, the domain of $::$ is Individual and the codomain is Type (a3). From (t1), (t2), entities are partitioned into individuals and types. We introduce the *specialization* relation between types ("$\sqsubseteq$") defining it in terms of necessary extensional inclusion (a4), i.e., inclusion of their instances. By means of (a4), 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 (a5).

a1 $\mathsf{Type}(x) \leftrightarrow \Diamond(\exists y(y :: x))$

a2 $\mathsf{Individual}(x) \leftrightarrow \Box(\neg\exists y(y :: x))$

a3 $x :: y \rightarrow \mathsf{Individual}(x) \wedge \mathsf{Type}(y)$

t1 $\mathsf{Individual}(x) \vee \mathsf{Type}(x)$

t2 $\neg\exists x(\mathsf{Individual}(x) \wedge \mathsf{Type}(x))$

a4 $x \sqsubseteq y \leftrightarrow \mathsf{Type}(x) \wedge \mathsf{Type}(y) \wedge \Box(\forall z(z :: x \rightarrow z :: y))$

t3 $x \sqsubseteq y \rightarrow (x \sqsubseteq x \wedge y \sqsubseteq y)$

t4 $x \sqsubseteq y \wedge y \sqsubseteq z \rightarrow x \sqsubseteq z$

a5 $\forall t_1, t_2, x((x :: t_1 \wedge x :: t_2 \wedge \neg(t_1 \sqsubseteq t_2) \wedge \neg(t_2 \sqsubseteq t_1)) \rightarrow (\exists t_3(t_1 \sqsubseteq t_3 \wedge t_2 \sqsubseteq t_3 \wedge x :: t_3) \vee \exists t_3(t_3 \sqsubseteq t_1 \wedge t_3 \sqsubseteq t_2 \wedge x :: t_3)))$

We implicitly define *rigidity* of types as rigid (a6), anti-rigid (a7) and semi-rigid (a8), concluding that every type is either one of the three ((t5) and (t6)) and rigid and semi-rigid types cannot specialize anti-rigid ones ((t7) and (t8)).

a6 $\mathsf{Rigid}(t) \leftrightarrow \mathsf{Type}(t) \wedge \forall x(\Diamond(x :: t) \rightarrow \Box(x :: t))$

a7 $\mathsf{AntiRigid}(t) \leftrightarrow \mathsf{Type}(t) \wedge \forall x(\Diamond(x :: t) \rightarrow \Diamond(\neg x :: t))$

a8 $\mathsf{SemiRigid}(t) \leftrightarrow \mathsf{Type}(t) \wedge \neg\mathsf{Rigid}(t) \wedge \neg\mathsf{AntiRigid}(t)$

t5 $\mathsf{Type}(t) \leftrightarrow \mathsf{Rigid}(t) \vee \mathsf{AntiRigid}(t) \vee \mathsf{SemiRigid}(t)$

t6 $\neg\exists x((\mathsf{Rigid}(x) \wedge \mathsf{AntiRigid}(x)) \vee (\mathsf{Rigid}(x) \wedge \mathsf{SemiRigid}(x)) \vee (\mathsf{SemiRigid}(x) \wedge \mathsf{AntiRigid}(x)))$

t7 $\neg\exists x, y(\mathsf{Rigid}(x) \wedge \mathsf{AntiRigid}(y) \wedge x \sqsubseteq y)$

t8 $\neg\exists x, y(\mathsf{SemiRigid}(x) \wedge \mathsf{AntiRigid}(y) \wedge x \sqsubseteq y)$

On *sortality*, our basic assumption is that every individual necessarily instantiates a kind (a9), and everything necessarily instantiates at most one kind (a10).[7] We implicitly define *sortals* as those types whose instances necessarily instantiate the same

[5] TPTP is available at http://www.tptp.org.

[6] For the formal specification, see https://github.com/nemo-ufes/ufo-types.

[7] These together represent the junction of: Quine's *"No entity without identity"* [30]; Macnamara's psychological claim 2 (no identity without a sortal); with the idea that if incompatible principles of identity apply to the same individual then it cannot be a viable entity (a *determinate individual*). This is because, under qualitative changes, different principles of identity can provide different answers to the question of whether that individual remains the same [2]. The later also corresponds to Macnamara's psychological claim 6: *"We cannot directly express identity across different kinds"* [13]. In accordance to this view, the philosopher David Wiggins defends that if an individual instantiates two sortals in the course of its history then there must be exactly one ultimate sortal of which both sortals are specialization [31]. This came to be known as *Wiggins' Thesis D*. Since this ultimate sortal must be rigid (given that a principle of identity must trace the identity of individuals in all possible situations), this unique ultimate rigid sortal specialized by other sortals that share its instances is our notion of *kind*.

kind (a11); while a *non-sortal* is a type that is necessarily not a sortal (a12). 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 a sortal that either specializes the non-sortal, or specializes a common non-sortal supertype (t16).

a9 $\mathsf{Individual}(x) \rightarrow \exists k(\mathsf{Kind}(k) \wedge \Box(x :: k))$

a10 $\mathsf{Kind}(k) \wedge x :: k \rightarrow \neg\Diamond(\exists z(\mathsf{Kind}(z) \wedge x :: z \wedge z \neq k))$

a11 $\mathsf{Sortal}(t) \leftrightarrow \mathsf{Type}(t) \wedge \exists k(\mathsf{Kind}(k) \wedge \Box(\forall x(x :: t \rightarrow x :: k)))$

a12 $\mathsf{NonSortal}(t) \leftrightarrow \mathsf{Type}(t) \wedge \neg\mathsf{Sortal}(t)$

t9 $\mathsf{Kind}(k) \rightarrow \mathsf{Rigid}(k)$

t10 $\mathsf{Kind}(x) \wedge \mathsf{Kind}(y) \wedge x \neq y \rightarrow \Box(\neg\exists z(z :: x \wedge z :: y))$

t11 $\mathsf{Kind}(x) \wedge \mathsf{Kind}(y) \wedge x \neq y \rightarrow (\neg(x \sqsubseteq y) \wedge \neg(y \sqsubseteq x))$

t12 $\mathsf{Kind}(t) \rightarrow \mathsf{Sortal}(t)$

t13 $\mathsf{Sortal}(x) \rightarrow \exists k(\mathsf{Kind}(k) \wedge x \sqsubseteq k)$

t14 $\neg\exists x, y, z(\mathsf{Kind}(y) \wedge \mathsf{Kind}(z) \wedge y \neq z \wedge x \sqsubseteq y \wedge x \sqsubseteq z)$

t15 $\neg\exists x, y(\mathsf{NonSortal}(x) \wedge \mathsf{Sortal}(y) \wedge x \sqsubseteq y)$

t16 $(\mathsf{NonSortal}(t) \wedge x :: t) \rightarrow (\exists s(\mathsf{Sortal}(s) \wedge s \sqsubseteq t \wedge x :: s) \vee \exists n, s(\mathsf{NonSortal}(n) \wedge \mathsf{Sortal}(s) \wedge s \sqsubseteq n \wedge t \sqsubseteq n \wedge x :: s))$

Regarding the leaves of the taxonomy of types according to their sortality and rigidity, kinds and subkinds are disjoint (a13), and together encompass all rigid sortals (a14). Phases and roles are disjoint (a15), and together encompass all anti-rigid sortals (a16). Semi-rigid sortals are those that are semi-rigid and sortal (a17). Categories are those types that are rigid and non-sortals (a18). Mixins are those types that are semi-rigid and non-sortals (a19). Phase-mixins and role-mixins are disjoint (a20), and together encompass all anti-rigid non-sortals (a21). Let $\mathcal{L}_T$ be the set of the leaf categories of the UFO taxonomy of types {Kind, SubKind, Role, Phase, SemiRigidSortal, RoleMixin, PhaseMixin, Category, Mixin}, it follows that these leaf categories are pairwise disjoint (t17) and complete (t18).

a13 $\neg\exists t(\mathsf{Kind}(t) \wedge \mathsf{SubKind}(t))$

a14 $\mathsf{Kind}(t) \vee \mathsf{SubKind}(t) \leftrightarrow \mathsf{Rigid}(t) \wedge \mathsf{Sortal}(t)$

a15 $\neg\exists t(\mathsf{Phase}(t) \wedge \mathsf{Role}(t))$

a16 $\mathsf{Phase}(t) \vee \mathsf{Role}(t) \leftrightarrow \mathsf{AntiRigid}(t) \wedge \mathsf{Sortal}(t)$

a17 $\mathsf{SemiRigidSortal}(t) \leftrightarrow \mathsf{SemiRigid}(t) \wedge \mathsf{Sortal}(t)$

a18 $\mathsf{Category}(t) \leftrightarrow \mathsf{Rigid}(t) \wedge \mathsf{NonSortal}(t)$

a19 $\mathsf{Mixin}(t) \leftrightarrow \mathsf{SemiRigid}(t) \wedge \mathsf{NonSortal}(t)$

a20 $\neg\exists t(\mathsf{PhaseMixin}(t) \wedge \mathsf{RoleMixin}(t))$

a21 $\mathsf{PhaseMixin}(t) \vee \mathsf{RoleMixin}(t) \leftrightarrow \mathsf{AntiRigid}(t) \wedge \mathsf{NonSortal}(t)$

t17 $\bigwedge_{i,j \in \mathcal{L}_T, i \neq j} i(t) \rightarrow \neg j(t)$

t18 $\mathsf{Type}(x) \leftrightarrow \bigvee_{i \in \mathcal{L}_T} i(x)$

On the UFO taxonomy of endurants, endurants are individuals (a22), and the Endurant type is partitioned into Substantial and Moment (a23), (a24). Moreover, the Moment type is partitioned into Relator and IntrinsicMoment (a25), (a26). Finally, the IntrinsicMoment type is partitioned into Mode and Quality (a27), (a28). Let $\mathcal{L}_e$ be the set of the leaf categories of endurants {Substantial, Relator, Mode, Quality}, it is a theorem that these leaf categories partition Endurant (t19), (a20).

a22 $\mathsf{Endurant}(x) \rightarrow \mathsf{Individual}(x)$

a23 $\mathsf{Substantial}(x) \vee \mathsf{Moment}(x) \leftrightarrow \mathsf{Endurant}(x)$

a24 $\neg\exists x(\mathsf{Substantial}(x) \wedge \mathsf{Moment}(x))$

a25 $\mathsf{Relator}(x) \vee \mathsf{IntrinsicMoment}(x) \leftrightarrow \mathsf{Moment}(x)$

a26 $\neg\exists x(\mathsf{Relator}(x) \wedge \mathsf{IntrinsicMoment}(x))$

a27 $\mathsf{Mode}(x) \vee \mathsf{Quality}(x) \leftrightarrow \mathsf{IntrinsicMoment}(x)$

a28 $\neg\exists x(\mathsf{Mode}(x) \wedge \mathsf{Quality}(x))$

t19 $\bigwedge_{i,j \in \mathcal{L}_e, i \neq j} i(t) \rightarrow \neg j(t)$

t20 $\mathsf{Endurant}(x) \leftrightarrow \mathsf{Substantial}(x) \vee \mathsf{Relator}(x) \vee \mathsf{Mode}(x) \vee \mathsf{Quality}(x)$

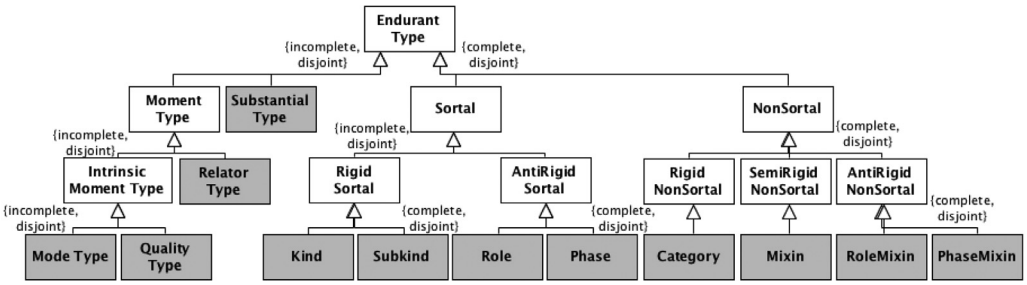


Fig. 3. Proposed taxonomy of endurant types in UFO.

We define a taxonomy of endurant types according to the ontological nature of their instances. Let $\mathcal{P}_E$ be the set of pairs {(EndurantType, Endurant); (SubstantialType, Substantial); (MomentType, Moment); (RelatorType, Relator); (ModeType, Mode); (QualityType, Quality)}. We implicitly define these types in the axiom schema (a29). It follows that these types are pairwise disjoint (t21).

a29 $\bigwedge_{(i,j)\in\mathcal{P}_E} i(t) \leftrightarrow \mathsf{Type}(t) \wedge \Box(\forall x(x :: t \rightarrow j(x)))$

t21 $\bigwedge_{i,j\in\{\mathsf{SubstantialType},\mathsf{RelatorType},\mathsf{ModeType},\mathsf{QualityType}\}, i\neq j} i(x) \rightarrow \neg j(x)$

Kinds are also specialized according to the ontological nature of their instances. Let $\mathcal{P}_K$ be the set of pairs {(SubstantialKind, SubstantialType); (RelatorKind, RelatorType); (ModeKind, ModeType); (QualityKind, QualityType)}. We implicitly define these kinds in the axiom schema (a30). 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 (a31). It follows that every endurant sortal is a type in $\mathcal{L}_{ES}$ = {SubstantialKind, RelatorKind, ModeKind, QualityKind, SubKind, Phase, Role, SemiRigidSortal} (a23); and that some sortals specialize specific kinds (t24).

a30 $\bigwedge_{(i,j)\in\mathcal{P}_K} i(t) \leftrightarrow j(t) \wedge \mathsf{Kind}(t)$

t22 $\Diamond(\exists k((\mathsf{SubstantialKind}(k) \vee \mathsf{RelatorKind}(k) \vee \mathsf{ModeKind}(k) \vee \mathsf{QualityKind}(k)) \wedge x :: k)) \rightarrow \mathsf{Endurant}(x)$

a31 $\mathsf{Endurant}(x) \rightarrow \Diamond(\exists k((\mathsf{SubstantialKind}(k) \vee \mathsf{RelatorKind}(k) \vee \mathsf{ModeKind}(k) \vee \mathsf{QualityKind}(k)) \wedge x :: k))$

t23 $\mathsf{EndurantType}(x) \wedge \mathsf{Sortal}(x) \rightarrow \bigvee_{i\in\mathcal{L}_{ES}} i(x)$

t24 $(\mathsf{Sortal}(t) \wedge (\mathsf{SubstantialType}(t) \vee \mathsf{RelatorType}(t) \vee \mathsf{ModeType}(t) \vee \mathsf{QualityType}(t))) \leftrightarrow \exists k((\mathsf{SubstantialKind}(k) \vee \mathsf{RelatorKind}(k) \vee \mathsf{ModeKind}(k) \vee \mathsf{QualityKind}(k)) \wedge t \sqsubseteq k)$

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

t25 $\bigwedge_{i,j\in\mathcal{L}_{ET}, i\neq j} i(t) \rightarrow \neg j(t)$

t26 $\mathsf{EndurantType}(x) \rightarrow \bigvee_{i\in\mathcal{L}_{ET}} i(x)$

The results of this section are summarized in Fig. 3. When contrasting this figure with Fig. 1, one can appreciate how in this new theory, the taxonomy reflecting the ontological *nature* of the entity being classified (e.g., whether a substantial, a mode, a relator) is orthogonal to the one reflecting meta-properties such as sortality, rigidity, etc. Although the formal characterization of this theory is generally defined for *Type*, from an ontological perspective, endurants are the natural bearers of modal properties [20]. As a consequence, the interpretation of modal notions such as rigidity and anti-rigidity only makes (ontological) sense when applied to *Endurant Types*. Finally, although a logically possible combination of meta-properties, the category of semi-rigid sortals has been excluded from our ontology, given that it seems to play no role in Conceptual Modeling [3], as also confirmed by the empirical analysis of OntoUML models.

4. A UML profile for modeling endurant taxonomic structures

OntoUML is an ODCM language that extends UML by defining a set of stereotypes in order to reflect UFO ontological distinctions into language constructs. Constructs decorated by OntoUML stereotypes carry precise semantics grounded in the underlying

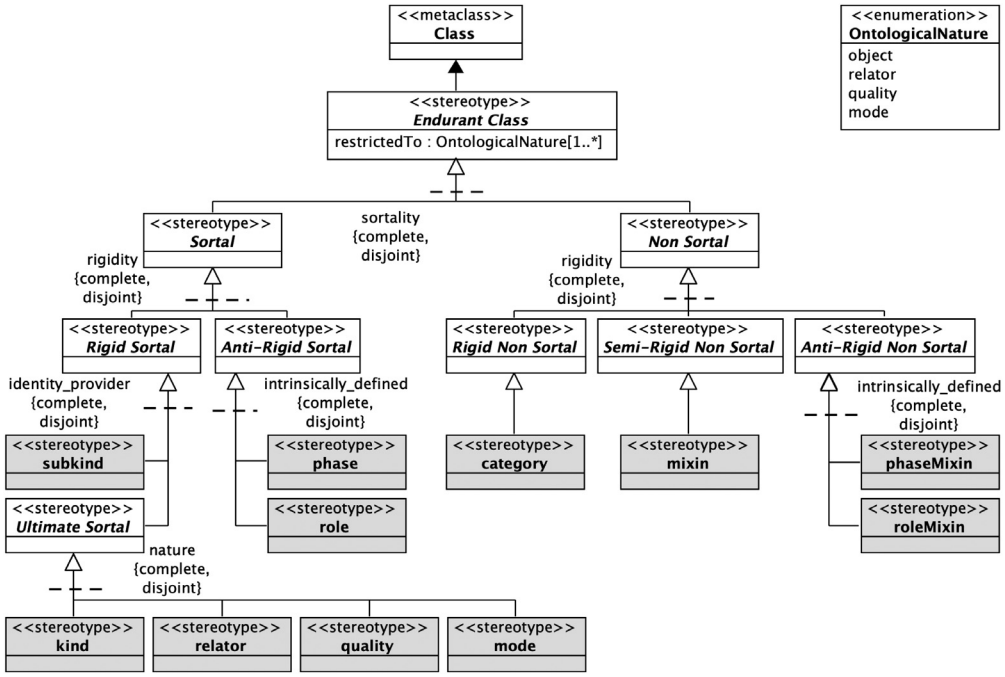


Fig. 4. UML profile for OntoUML 2.

ontology. Additionally, a number of *semantically motivated syntactic constraints* [32] govern OntoUML models driving them to conform to UFO. This combination of stereotypes and constraints enforce this conformance, making every valid OntoUML model compliant to UFO. In Section 4.1, we provide a new UML profile (*lightweight extension*) for OntoUML (henceforth termed OntoUML 2) that reflects the taxonomy of endurant types previously discussed and presented in Fig. 3. Later, in Section 4.2, we further develop OntoUML's *semantically motivated syntactic constraints* with additional representation in OCL.

4.1. Detailing the profile of OntoUML 2

The UML profile mechanism allows users to define new syntactical elements by extending the language's metamodel with stereotypes. In other words, to define a new class element the user extends the corresponding metaclass in UML with a stereotype that becomes available to decorate classes in user models. In Fig. 4, we employ this mechanism to define a profile that reflects the UFO taxonomy of Fig. 3 such that every class decorated with one of these stereotypes instantiates corresponding modal and ontological properties. The OntoUML profile presented here enables the representation of endurant types as UML classes. Notice that only the stereotypes in gray constitute valid elements in OntoUML, while the stereotypes in white are defined as *abstract* (i.e., classifiers that are under-specified and cannot have direct instances).

The profile of Fig. 4 focus on the classification of endurant types based on their modal properties (i.e., sortal and non-sortal, rigid, anti-rigid and semi-rigid, and so on) promoting a consistent set of stereotypes with traditional OntoUML (**«kind»**, **«relator»**, **«mode»**, **«quality»**, **«subkind»**, **«role»**, **«phase»**, **«category»**, **«roleMixin»**, **«phaseMixin»**, and **«mixin»**). To enable stereotypes to represent not only modal properties of classes but also the ontological nature of their instances, we rely on UFO's notion of *kinds* as *ultimate sortals*. For each ontological nature in UFO (i.e., substantial, relator, mode, and quality), we define in OntoUML 2 a single ultimate sortal stereotype – **«kind»**, **«relator»**, **«mode»**, and **«quality»** – where **«kind»** is reserved for ultimate sortals whose instances are objects (i.e., substantials).[8] The application of these stereotypes is exemplified in Fig. 5 and listed below.

- **«kind»**: decorates classes that represent ultimate sortals whose instances are regular objects. Examples of kinds include Person, House, and Organization.
- **«relator»**: decorates classes that represent ultimate sortals whose instances are relators. Examples of relators include Marriage, Insurance Contract, Employment, and Event Plan.
- **«quality»**: decorates classes that represent ultimate sortals whose instances are qualities. Examples of qualities include Weight, Height, BMI (body mass index), and Geometry.

[8] In OntoUML, the stereotype **«kind»** has been traditionally used to represent kinds of objects (i.e., ultimate sortals whose instances are objects), a terminology that we keep in this profile.

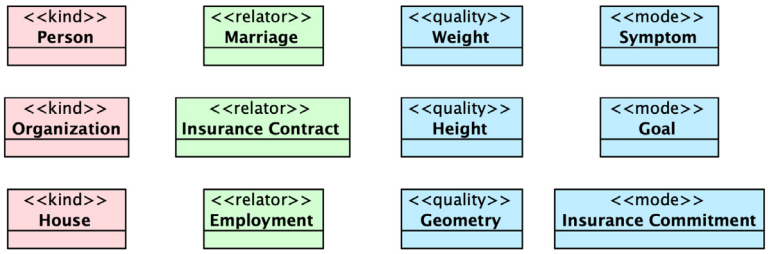


Fig. 5. Examples of *ultimate sortals* represented in OntoUML.

- **«mode»**: decorates classes that represent ultimate sortals whose instances are modes. Examples of modes include Goal, Symptom, Insurance Commitment, and Employment Commitment.

From (a9), (a13), (a14), and (a22) we have that every endurant must instantiate a unique ultimate sortal. Therefore, we can rely on these stereotypes to represent the ontological nature of their extensions knowing that every other class of endurants will be involved with ultimate sortals in their taxonomies, allowing for an inference of the possible natures of these other classes' instances. In OntoUML, the stereotypes listed above represent now rigid classes that are both the identity provider of their instances and the identifier of their ontological nature.

From (t24), every class representing an endurant sortal that is not a kind (i.e., **«subkind»**, **«phase»**, or **«role»**) specializes a unique ultimate sortal. These sortal classes carry the ultimate sortal's identity principle to their instances and represent the same nature it identifies. We present examples of sortal classes in Fig. 6 followed by the stereotypes' definitions. Notice that these examples are not intended to defend the particular modeling choices therein, but rather to elucidate language application. In order to improve model readability, we also use a color-coding to classes to represent the nature of their possible instances where types of objects are represented by classes in red, types of relators by classes in green, and types of mode or qualities by classes in blue.

- **«subkind»**: decorates classes that represent rigid sortals whose instances have a common ontological nature and follow a unique identity principle. Examples of subkinds include Nonprofit Organization, Commercial Organization, Life Insurance, and House Insurance.
- **«phase»**: decorates classes that represent (externally) independent anti-rigid sortals whose instances have a common ontological nature and follow a unique identity principle. Examples of phases include Child, Teenager, and Adult, Short-Term Relationship and Long-Term Relationship, as well as Underweight, Normal Weight and Overweight.
- **«role»**: decorates classes that represent externally dependent anti-rigid sortals whose instances have a common ontological nature and follow a unique identity principle. Examples of roles include Business Provider, Spouse, Significant Other, Foreign Marriage, and Stable Civil Partnership.

Whereas in traditional OntoUML stereotypes of **«subkind»**, **«phase»** and **«role»** would be restricted to substantials, in OntoUML 2 classes decorated with these stereotypes must inherit from a unique ultimate sortal both the identity principle and the restriction to a single ontological nature of endurants. For instance, in Fig. 6, Organization is specialized into the two rigid sortal types Nonprofit Organization and Commercial Organization, which constitute examples of subkinds. Likewise, Life Insurance, House Insurance, and Employment Insurance stand as examples of subkinds but that apply to relators rather substantials as we can infer from their specialization from Insurance Contract. We apply this same inference to anti-rigid sortals, allowing us to represent Business Provider, Spouse, Child, and Adult as roles and phases of substantials, Foreign Marriage, Short-Term Relationship, and Long-Term Relationship as roles and phases of relators, and Underweight, Normal Weight, and Overweight as phases of qualities.

These capabilities of OntoUML 2 allow us to enhance the expressivity in hierarchies of relators, modes, and qualities recognizing that these types can be defined in terms of:

- *relational properties*, in the case of Foreign Marriage for domains involving immigration;
- *dynamic internal properties*, in the case of BMI which characterizes a person's physical characteristics according to changing weight and height;

We can even acknowledge that, in rigid hierarchies, a unique type provides the identity principle for an extension of moments. E.g., Insurance Contract being the natural kind for all instances of Life Insurance, House Insurance, and Employment Insurance.

To non-sortals, on the other hand, the impact of the new OntoUML profile is the opposite. Stereotypes that decorated non-sortal classes (i.e., **«category»**, **«phaseMixin»**, **«roleMixin»**, and **«mixin»**) represent more general classes that apply to endurants following distinct identity principles or, in other words, endurants that instantiate different ultimate sortals. Fig. 7 presents examples of non-sortal classes whose stereotypes are defined in the list below:

- **«category»**: decorates classes that represent rigid non-sortals whose instances may follow different identity principles and are not constrained to a specific ontological nature. Examples of category include Agreement, Commitment, and Legal System.

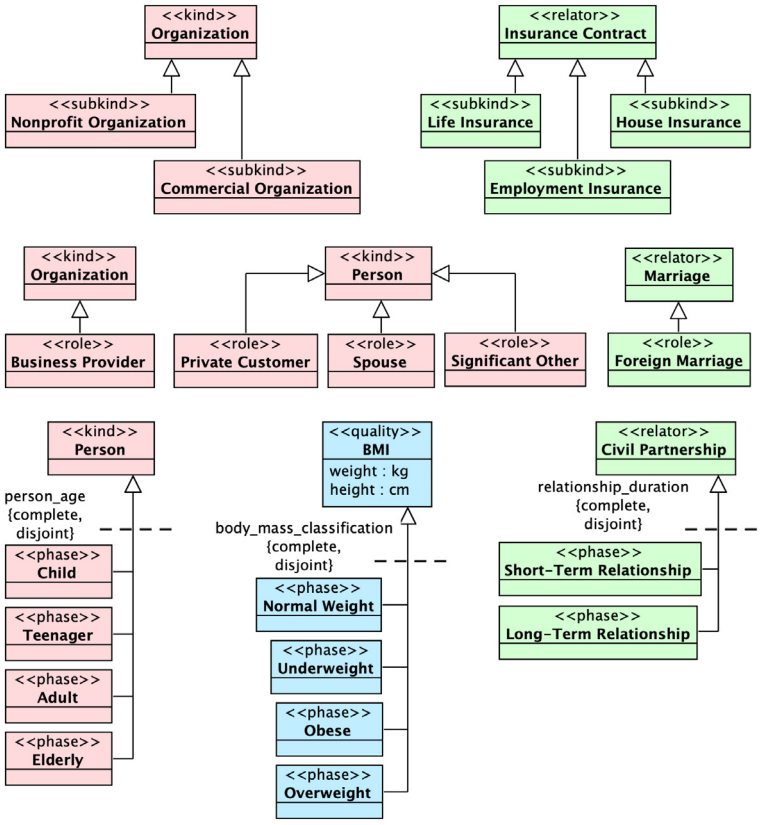


Fig. 6. Examples of sortal classes represented in OntoUML.

- **«phaseMixin»**: decorates classes that represent independent anti-rigid non-sortals whose instances may follow different identity principles and are not constrained to a specific ontological nature. Examples of phase mixin include Fulfilled Commitment, Unfulfilled Commitment, and Broken Commitment.
- **«roleMixin»**: decorates classes that represent externally-dependent anti-rigid non-sortals whose instances may follow different identity principles and are not constrained to a specific ontological nature. Examples of role mixin include Provider, Customer, and Insured Item.
- **«mixin»**: decorates classes that represent semi-rigid non-sortals whose instances may follow different identity principles and are not constrained to a specific ontological nature. Examples of mixin include Legally Recognized Civil Partnership.

From the examples of Fig. 7, traditional OntoUML would only be able to represent as non-sortals types of substantials, for instance, Provider, Customer and Legal System. In OntoUML 2, however, we may represent non-sortals such as Agreement, which is a general category of relators whose instances are agreements of several kinds (e.g., insurance contracts, marriages, and civil partnerships). Involved Part is a non-sortal role (i.e., role mixin) that classifies entities involved in some agreement. Involved parts that bear some Commitment within an agreement are classified as Party. An example of non-sortal phases (i.e., phase mixins) are the phase of Commitment, Fulfilled Commitment, Unfulfilled Commitment and Broken Commitment. Finally, Legally Recognized Conjugal Relationship is a mixin that classifies endurants that are involved in some Legal Recognition. Instances of it include both entities that are necessarily classified as such (e.g., marriages), as well as entities that are contingently classified (e.g., civil partnerships that are long-term).

Differently from sortal classes, non-sortal classes do not explicitly state the ontological nature of its instances through their stereotypes in this profile. Thus, in addition to classifying endurants of distinct identity principles, they can also classify endurants of different natures. This feature allows the modeler to capture types such as Insurance Item. Insurance Item is a sort of Involved Part that classifies endurants insured by some insurance agreement. This example of role mixin includes as instances both substantials (e.g., cars, houses, machines) and moments (e.g., an employment contract), mixing instances of different basic ontological categories.

The nature of non-sortals can be inferred in complete ontologies from the kinds of their instances, examining specialization hierarchies. However, with the addition of the restrictedTo tagged value in the profile (see Fig. 4) the natures of a non-sortal's instances can be explicitly represented, without relying on inference mechanisms and improving clarity in incomplete ontologies (as in core and reference ontologies). For instance, UML allows the usage of constraints for assignment of tagged values, which we have employed in Fig. 8. Here, it is explicit that the instances of Involved Part and Insured Item (color-coded in gray because

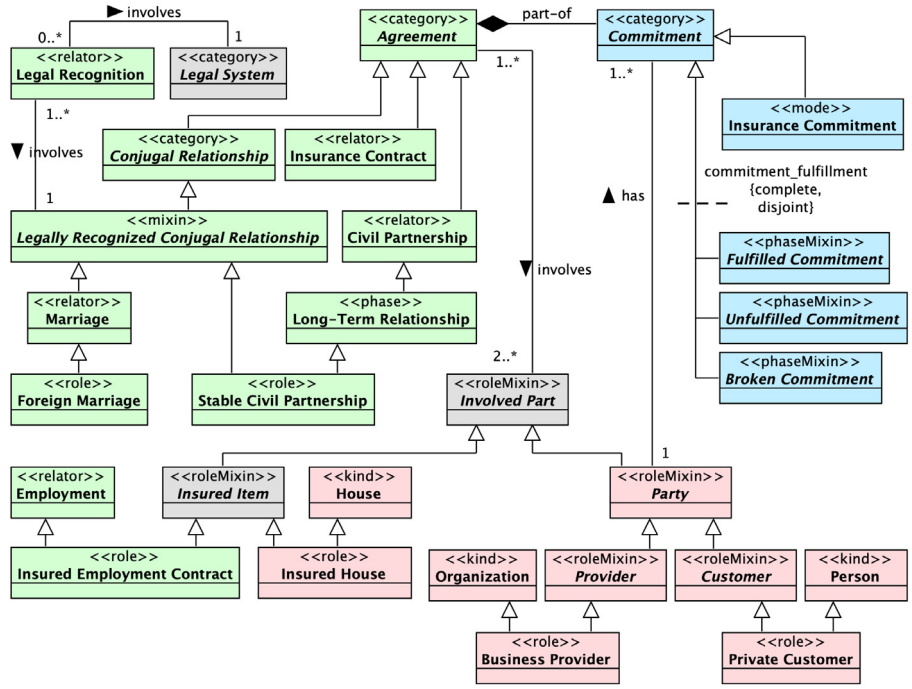


Fig. 7. Examples of non-sortal classes.

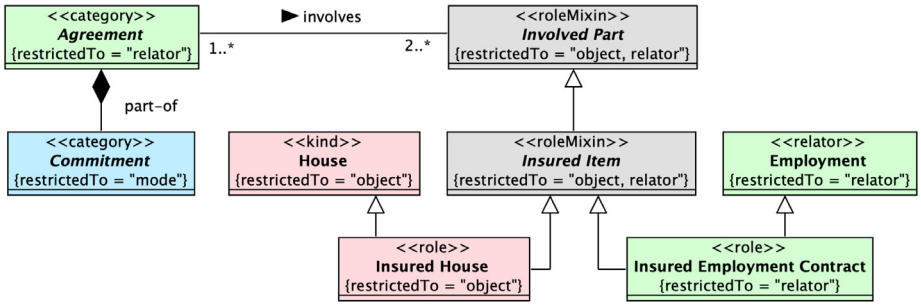


Fig. 8. Examples of non-sortal classes with restrictedTo.

they are not restricted to a single ontological nature) must be either objects or relators. Moreover, even without the presence of additional ultimate sortals in this diagram, it is clear to the reader that instances of Agreement and Commitment are restricted to relators and modes, respectively. This feature, even though of optional representation in diagrams, improves model readability across views as well and can also be used in diagrams of sortal classes in the absence of their kinds.

4.2. OntoUML 2 constraints

In this section, we list all the constraints of this profile of OntoUML 2 that emerge from the axiomatization in Section 3. The constraints discussed here, also referred to as *semantically motivated syntactic constraints*, are limited to those related to taxonomies of endurant types. The list of constraints is presented in both natural language (with reference to their grounding theorems and axioms) as well as in OCL. The OCL version of our constraints was implemented and tested on the Papyrus UML tool [33].

Constraint: onlyoneontoumlstereotype. From (t25) and (t26), every class representing an endurant type must be decorated with exactly one stereotype from the list: **«kind»**, **«relator»**, **«mode»**, **«quality»**, **«subkind»**, **«role»**, **«phase»**, **«category»**, **«mixin»**, **«roleMixin»**, **«phaseMixin»**. Semi-rigid sortals are excluded from the profile (see Section 3).

```
context EndurantSortal inv onlyOneOntoUMLStereotype:
   base_Classifier.getAppliedStereotypes()->select(
      name='kind' or name='relator' or name='mode' or
      name='quality' or name='phase' or name='role' or
```

Fig. 9. Counterexamples of onlyOneOntoUMLStereotype constraint.

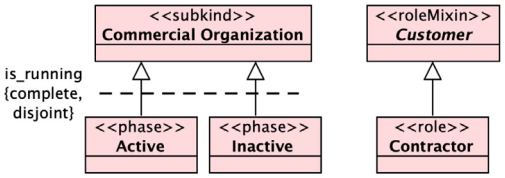


Fig. 10. Counterexamples of sortalMustSpecializeUltimateSortal constraint.

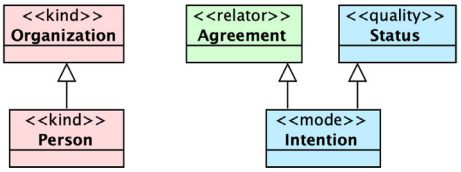


Fig. 11. Counterexamples of ultimateSortalCantSpecializeAnother constraint.

```
        name='subkind' or name='phaseMixin' or name='roleMixin' or
        name='category' or name='mixin'
    )->size()=1
```

Listing 1: Constraint onlyOneOntoUMLStereotype represented in OCL.

Fig. 9 presents a counterexample that violates the constraint onlyOneOntoUMLStereotype by applying multiple stereotypes to a single class element.

Constraint: sortalmustspecializeultimatesortal. From (t24), every class representing an endurant sortal that is not a kind (including **«subkind»**, **«role»**, **«phase»**) must specialize a class decorated with a stereotype **«kind»**, **«relator»**, **«mode»**, or **«quality»**.

```
context Sortal inv sortalMustSpecializeUltimateSortal:
    self.oclIsKindOf(UltimateSortal)
    or (
        base_Class.allParents()->exists(
            not oclAsType(Class).extension_EndurantClass
                .oclAsType(UltimateSortal).oclIsInvalid()
        )
    )
```

Listing 2: Constraint sortalMustSpecializeUltimateSortal represented in OCL.

Fig. 10 presents a model that, if complete, contains sortal classes that lack specialization of some ultimate sortal, violating the constraint sortalMustSpecializeUltimateSortal.

Constraint: ultimatesortalcantspecializeanother. From (t11), a class representing a kind cannot specialize another kind.

```
context UltimateSortal inv ultimateSortalCantSpecializeAnother:
    not base_Class.allParents()->exists(
        not oclAsType(Class).extension_EndurantClass
            .oclAsType(UltimateSortal).oclIsInvalid()
    )
```

Listing 3: Constraint ultimateSortalCantSpecializeAnother represented in OCL.

Fig. 11 presents counterexamples that violate the constraint ultimateSortalCantSpecializeAnother by Person specializing Organization, and Intention specializing Agreement and Status.

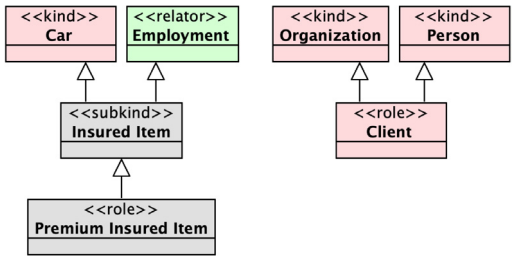


Fig. 12. Counterexamples of cantSpecializeMoreThanOneUltimateSortal constraint.

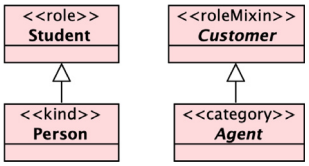


Fig. 13. Counterexamples of rigidSortalCantSpecializeAntiRigid and rigidNonSortalCantSpecializeAntiRigid constraints.

Constraint: cantspecializemorethanoneultimatesortal. From (t14), a class cannot specialize more than one kind.

```
context Sortal inv cantSpecializeMoreThanOneUltimateSortal:
    base_Class.allParents()->select(
        not oclAsType(Class).extension_EndurantClass
            .oclAsType(UltimateSortal).oclIsInvalid()
    )->size()<=1
```

Listing 4: Constraint cantSpecializeMoreThanOneUltimateSortal represented in OCL.

Fig. 12 presents counterexamples that violate the constraint cantSpecializeMoreThanOneUltimateSortal by Client specializing Person and Organization, Insured Item specializing Car and Agreement, as well as Premium Insured Item specializing (indirectly) Car and Agreement.

Constraints: rigidsortalcantspecializeantirigid and rigidnonsortalcantspecializeantirigid. From (t7), a class representing a rigid type (**«kind»**, **«relator»**, **«mode»**, **«quality»**, **«subkind»**, **«category»**) cannot specialize a class representing an anti-rigid type (**«role»**, **«phase»**, **«roleMixin»**, **«phaseMixin»**).

```
context RigidSortal inv rigidSortalCantSpecializeAntiRigid:
    not base_Class.allParents()->exists(
        not oclAsType(Class).extension_EndurantClass
            .oclAsType(AntiRigidSortal).oclIsInvalid()
        or
        not oclAsType(Class).extension_EndurantClass
            .oclAsType(AntiRigidNonSortal).oclIsInvalid()
    )
```

Listing 5: Constraint rigidSortalCantSpecializeAntiRigid represented in OCL.

```
context RigidNonSortal inv rigidNonSortalCantSpecializeAntiRigid:
    not base_Class.allParents()->exists(
        not oclAsType(Class).extension_EndurantClass
            .oclAsType(AntiRigidSortal).oclIsInvalid()
        or
        not oclAsType(Class).extension_EndurantClass
            .oclAsType(AntiRigidNonSortal).oclIsInvalid()
    )
```

Listing 6: Constraint rigidNonSortalCantSpecializeAntiRigid represented in OCL.

Fig. 13 presents counterexamples that violate the constraints rigidSortalCantSpecializeAntiRigid, by having Person specializing Student, and rigidNonSortalCantSpecializeAntiRigid, by having Agent specializing Customer.

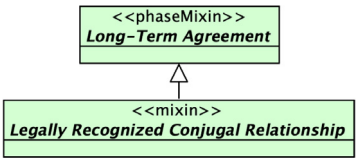


Fig. 14. Counterexamples of semiRigidCantSpecializeAntiRigid constraint.

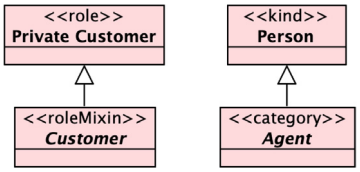


Fig. 15. Counterexamples of nonSortalCantSpecializeSortal constraint.

Constraint: semirigidcantspecializeantirigid. From (t8), a class representing a semi-rigid type (**«mixin»**) cannot specialize a class representing an anti-rigid type (**«role»**, **«phase»**, **«roleMixin»**, **«phaseMixin»**).

```
context SemiRigidNonSortal inv semiRigidCantSpecializeAntiRigid:
  not base_Class.allParents()->exists(
    not oclAsType(Class).extension_EndurantClass.
          oclAsType(AntiRigidSortal).oclIsInvalid() or
    not oclAsType(Class).extension_EndurantClass.
          oclAsType(AntiRigidNonSortal).oclIsInvalid()
  )
```

Listing 7: Constraint semiRigidCantSpecializeAntiRigid represented in OCL.

Fig. 14 presents counterexamples that violate the constraint semiRigidCantSpecializeAntiRigid by Legally Recognized Conjugal Relationship specializing Long-Term Agreement.

Constraint: nonsortalcantspecializesortal. From (t15), a class representing a non-sortal (**«category»**, **«mixin»**, **«roleMixin»**, **«phaseMixin»**) cannot specialize a class representing a sortal one (**«kind»**, **«relator»**, **«mode»**, **«quality»**, **«subkind»**, **«role»**, or **«phase»**).

```
context NonSortal inv nonSortalCantSpecializeSortal:
  not base_Class.allParents()->exists(
    not oclAsType(Class).extension_EndurantClass.oclAsType(Sortal).oclIsInvalid()
  )
```

Listing 8: Constraint nonSortalCantSpecializeSortal represented in OCL.

Fig. 15 presents counterexamples that violate the constraint nonSortalCantSpecializeSortal by Customer specializing Private Customer and Agent specializing Person.

Constraint: nonsortalmusthavesortalspecialization. From (t16), given a non-sortal N, there must be a sortal S that specializes N, or specializes a non-sortal supertype common to both N and S.

```
context NonSortal inv nonSortalMustHaveSortalSpecialization:
    -- there is a sortal that has the non-sortal as parent
    Sortal.allInstances()->exists(
        base_Class.allParents()->includes(self.base_Class)
    )
    or
    -- there is a sortal with a parent x that is also parent of the non-sortal
    Sortal.allInstances()->exists(
        base_Class.allParents()->exists(
            x | self.base_Class.allParents()->includes(x)
        )
    )
```

Listing 9: Constraint nonSortalMustHaveSortalSpecialization represented in OCL.

5. A lightweight implementation of the unified foundational ontology in OWL

Beyond the OntoUML profile, the redesigned UFO theory presented in Section 3 also drives the implementation of an OWL ontology dubbed gUFO.[9] This lightweight implementation of UFO allows users to harness UFO's conceptualization in Semantic Web OWL 2 DL [34] applications, enabling the exploration of the technologies available in this field. In OWL, gUFO allows for the specification of both models and instances, the usage of reasoners and querying languages (e.g., SPARQL), integration with triple stores, and integration to other ontologies in the Semantic Web for example. The "g" in gUFO stands for *gentle*, while the word "gufo" itself is the Italian translation for owl, the animal.

The implementation of gUFO in OWL reproduces, with the limitations imposed by the expressiveness of this language, the concepts presented in Section 3. Its users are meant to directly extend it by instantiating and specializing its classes and properties. This approach differs from that of OntoUML in the sense that no syntactical rules are added to OWL and users have full control on how to interact with the ontology.

gUFO is organized around two hierarchies that reflect that depicted in Fig. 3. One for endurants and one for endurant types. The hierarchy of endurants captures specializations of Endurant according to distinct ontological natures partitioning endurants into Object (i.e., Substantial), Relator, Mode, or Quality. In order to define a class of endurants in a domain ontology identifying the ontological nature of its instances, the user must specify a rdfs:subClassOf statement towards one of the aforementioned specializations of Endurant. For example, the Turtle fragment in Listing 10 defines that Person and PrivateCustomer are classes of objects, the former through direct specialization of Object and the latter through its indirect specialization via Person.

```
PREFIX gufo: <http://purl.org/nemo/gufo#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

:Person rdfs:subClassOf gufo:Object .
:PrivateCustomer rdfs:subClassOf :Person .
```

Listing 10: Example of classes specializing Endurant in gUFO-based UFO ontologies.

The hierarchy of endurant types captures specializations of EndurantType according to their modal properties (e.g., sortality and rigidity) partitioning them into Kind (i.e., ultimate sortals), SubKind, Role, Phase, Category, RoleMixin, PhaseMixin, or Mixin. In order to represent the modal properties of a class of endurants in a domain ontology, the user must specify a rdfs:type statement towards one of the aforementioned specializations of EndurantType. For example, in Listing 11 the class Person represents an ultimate sortal and, thus, instantiates Kind with a rdf:type statement. Likewise, PrivateCustomer represents a role (i.e., an externally-dependent anti-rigid sortal), as captured by the instantiation of Role.

```
PREFIX gufo: <http://purl.org/nemo/gufo#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

:Person rdf:type gufo:Kind;
   rdfs:subClassOf gufo:Object .
:PrivateCustomer rdf:type gufo:Role;
   rdfs:subClassOf :Person .
```

Listing 11: Example of classes instantiating EndurantType in gUFO-based UFO ontologies.

To create gUFO-based domain ontologies, users need to specialize and instantiate its concepts and relations. However, to keep their ontologies sound in regard to UFO, they must make sure that their ontologies comply with the constraints presented in Section 4.2. In the following paragraphs, we provide a translation of the OntoUML constraints, presented earlier in OCL, to SPARQL queries designed to detect anti-patterns in the user's ontologies caused by the violation of these constraints.[10] These queries should improve ODCM in gUFO-based ontologies even in a context that lacks the support of a constraining mechanism such as OCL.

Constraint: onlyoneontoumlstereotype. Listing 12 presents a SPARQL query for detecting classes violating the constraint onlyOneOntoUMLStereotype.

This constraint ensures the adequate extension of gUFO via specialization of leaf classes in the hierarchy of gufo:Endurant and instantiation of leaf classes in the hierarchy of gufo:EndurantType. It is meant to verify that the class in the user's ontology adequately represents its ontological nature (through specialization) and modal properties (through instantiation).

```
PREFIX gufo: <http://purl.org/nemo/gufo#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

[9] http://purl.org/nemo/doc/gufo.

[10] In order to represent UFO's types of types in gUFO, our representation strategy employs OWL 2 punning, i.e., we allow also for classes to appear as instances of other classes. We then express as *violation-detection queries* in SPARQL, as those level-crossing integrity constraints that cannot be directly captured in OWL 2 DL.

```
select distinct ?type where { {
       select ?type (count(?type) as ?n) where {
           { ?type rdf:type gufo:Category . }
           union { ?type rdf:type gufo:RoleMixin . }
           union { ?type rdf:type gufo:PhaseMixin . }
           union { ?type rdf:type gufo:Mixin . }
           union { ?type rdf:type gufo:Kind . }
           union { ?type rdf:type gufo:SubKind . }
           union { ?type rdf:type gufo:Role . }
           union { ?type rdf:type gufo:Phase . }
       } group by ?type
    } filter (?n > 1)
}

select distinct ?type where { {
       select ?type (count(?type) as ?n) where {
           { ?type rdfs:subClassOf gufo:Object . }
           union { ?type rdfs:subClassOf gufo:Relator . }
           union { ?type rdfs:subClassOf gufo:Quality . }
           union { ?type rdfs:subClassOf gufo:ExtrinsicMode . }
           union { ?type rdfs:subClassOf gufo:IntrinsicMode . }
       } group by ?type
    } filter (?n > 0)
}
```

Listing 12: SPARQL queries for detecting violations of the constraint onlyOneOntoUMLStereotype in gUFO ontologies.

Constraint: sortalmustspecializeultimatesortal. Listing 13 presents a SPARQL query for detecting entities violating the constraint sortalMustSpecializeUltimateSortal.

```
PREFIX gufo: <http://purl.org/nemo/gufo#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

select distinct ?sortal where {
    { ?sortal rdf:type gufo:SubKind . }
    union { ?sortal rdf:type gufo:Phase . }
    union { ?sortal rdf:type gufo:Role . }
    filter not exists {
        ?sortal rdfs:subClassOf* ?ultimateSortal .
        ?ultimateSortal rdf:type gufo:Kind .
    }
}
```

Listing 13: SPARQL query for detecting violations of the constraint sortalMustSpecializeUltimateSortal in gUFO ontologies.

Constraint: ultimatesortalcantspecializeanother. Listing 14 presents a SPARQL query for detecting entities violating the constraint ultimateSortalCantSpecializeAnother.

```
PREFIX gufo: <http://purl.org/nemo/gufo#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

select distinct ?ultimateSortal where {
    ?ultimateSortal rdf:type gufo:Kind .
    filter exists {
        ?ultimateSortal rdfs:subClassOf+/rdf:type gufo:Kind .
    }
}
```

Listing 14: SPARQL query for detecting violations of the constraint ultimateSortalCantSpecializeAnother in gUFO ontologies.

Constraint: cantspecializemorethanoneultimatesortal. Listing 15 presents a SPARQL query for detecting entities violating the constraint cantSpecializeMoreThanOneUltimateSortal.

```
PREFIX gufo: <http://purl.org/nemo/gufo#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

select distinct ?sortal where { {
       select ?sortal (count(?sortal) as ?n)
       where {
           { ?sortal rdf:type gufo:SubKind . }
           union { ?sortal rdf:type gufo:Phase . }
           union { ?sortal rdf:type gufo:Role . }
           ?sortal rdfs:subClassOf+ ?ultimateSortal .
           ?ultimateSortal rdf:type gufo:Kind .
       } group by ?sortal
   } filter(?n > 1)
}
```

Listing 15: SPARQL query for detecting violations of the constraint cantSpecializeMoreThanOneUltimateSortal in gUFO ontologies.

Constraints: rigidsortalcantspecializeantirigid, rigidnonsortalcantspecializeantirigid, and semirigidcantspecializeantirigid. Listing 16 presents a SPARQL query for detecting entities violating the constraint rigidSortalCantSpecializeAntiRigid, rigidNonSortalCantSpecializeAntiRigid, and semiRigidCantSpecializeAntiRigid.

```
PREFIX gufo: <http://purl.org/nemo/gufo#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

select distinct ?type where {
    { ?type rdf:type/rdfs:subClassOf* gufo:RigidType . }
    union { ?type rdf:type/rdfs:subClassOf* gufo:SemiRigidType . }
    ?type rdfs:subClassOf+ ?antiRigidType .
    ?antiRigidType rdf:type/rdfs:subClassOf* gufo:AntiRigidType .
}
```

Listing 16: SPARQL query for detecting violations of the constraints rigidSortalCantSpecializeAntiRigid, rigidNonSortalCantSpecializeAntiRigid, and semiRigidCantSpecializeAntiRigid in gUFO ontologies.

Constraint: nonsortalcantspecializesortal. Listing 17 presents a SPARQL query for detecting entities violating the constraint nonSortalCantSpecializeSortal.

```
PREFIX gufo: <http://purl.org/nemo/gufo#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

select distinct ?nonSortal where {
    ?nonSortal rdf:type/rdfs:subClassOf* gufo:NonSortal .
    ?nonSortal rdfs:subClassOf+ ?sortal .
    ?sortal rdf:type/rdfs:subClassOf* gufo:Sortal .
}
```

Listing 17: SPARQL query for detecting violations of the constraint nonSortalCantSpecializeSortal in gUFO ontologies.

Constraint: nonsortalmusthavesortalspecialization. Listing 18 presents a SPARQL query for detecting entities violating the constraint nonSortalMustHaveSortalSpecialization.

```
PREFIX gufo: <http://purl.org/nemo/gufo#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

select distinct ?nonSortal where {
    ?nonSortal rdf:type/rdfs:subClassOf* gufo:NonSortal .
    filter not exists {
        {
            ?sortal rdf:type/rdfs:subClassOf* gufo:Sortal .
            ?sortal rdfs:subClassOf* ?nonSortal .
        }
```

```
        union
        {
            ?otherNonSortal rdf:type/rdfs:subClassOf* gufo:NonSortal .
            ?sortal rdf:type/rdfs:subClassOf* gufo:Sortal .
            ?sortal rdfs:subClassOf+ ?otherNonSortal .
            ?nonSortal rdfs:subClassOf+ ?otherNonSortal .
        }
    }
}
```

Listing 18: SPARQL query for detecting violations of the constraint nonSortalMustHaveSortalSpecialization in gUFO ontologies.

The implementation of UFO in gUFO requires a distinct approach to modeling. Whereas OntoUML makes UFO concepts available as syntactical elements, gUFO reflects UFO concepts directly in an ontology making them available to the user at the semantical level. While the OntoUML profile restricts the interactions of modelers with the grounding theory to the set of available stereotypes, the user of gUFO directly specializes and instantiates UFO concepts when building models.

In other words, the user of gUFO bears greater responsibility when it comes to building consistent UFO-based ontologies while having more freedom to directly interact with UFO. This also highlights, in comparison, the design philosophy of OntoUML as an ontology-driven conceptual modeling language as, instead of working with triples (in OWL) or regular classes (in bare UML) as syntactical elements, the modeler has in OntoUML at his/her disposal more refined and well-founded elements (e.g., kinds, roles, and phases) to perform conceptual modeling. The goal of OntoUML is to shift certain responsibilities in conceptual modeling from the modeler to the language and its constraints, while the goal of gUFO is to make UFO available for the Semantic Web and its applications.

6. Tool support

Along with this new OntoUML profile, the revision of the language requires the development of adequate tooling support for ODCM. Even considering the capabilities enabled by UML's profiling mechanism and OCL constraint language, UML CASE tools are limited in regard to how much they can support the user during conceptual modeling tasks. In that spirit, we have developed a new set of tools designed to take advantage of OntoUML's rich semantics and provide support to modeler beyond the capabilities of regular UML CASE tools.

For modelers, we have developed the ontouml-vp-plugin,[11] a plug-in for the Visual Paradigm UML CASE tool[12] that enables OntoUML 2 modeling by installing the necessary stereotypes and tagged values and adding features such as a personalized user interface, smart coloring of diagrams, auto-completion of elements properties, partial and complete model verification, and model transformation (including transformation to OWL as gUFO ontologies); see Fig. 16 For instance, Fig. 17 presents a screenshot of ontouml-vp-plugin running on Visual Paradigm and displaying to the user the violation of an OntoUML constraint. The model verification feature is enabled by a remote service that notifies the plugin of detected violations, their severity (e.g., "error" or "warning"), and their cause. The plugin, in its turn, displays eventual violation messages to the modeler, showing offending model elements upon request.

For developers, the mechanisms of model serialization and verification/manipulation are made available through the ontouml-schema[13] and ontouml-js[14] modules, respectively. These modules are JavaScript projects made openly available to developers interested in designing OntoUML-based solutions.

7. Psychological evidence

As previously discussed, our theory was also inspired by a list of psychological claims proposed by the cognitive psychologist John Macnamara in [13]. The proposed psychological claims are related to the cognitive interpretation of linguistic expressions. In that article, Macnamara defends the position that there is a logic underlying the fact that we can understand certain propositions, and the proposed set of psychological claims points to a class of logics that takes cognizance of this fact. This position, which is also supported by, for instance, [35–37] is analogous to the one advocated by Chomsky in defense of his notion of a Universal Grammar [38–40]. Chomsky is the proponent of the theory that states that the reason why we can learn a natural language is due to the existence of a *mental grammar*, a *linguistic competence* that is nature-supplied, uniform across individuals and complete in each one. According to this view, there is a close fit between the mind's linguistic properties and properties of natural languages and, hence, natural languages have the properties they do because they can be recognized and manipulated by infants without the meta-linguistic support that is available to second-language learners. Therefore, for Chomsky, a grammar for a particular language is *descriptively adequate* if it correctly describes its object, namely the linguistic intuition of the native speaker.

In the same spirit, a number of cognitive scientists (see, for example, [13,35]), have proposed a theory of *logical competence* based on the notion of a *Language of Though* [41]. For Fodor [41], the reason we can learn the *meaning* of natural language symbols

[11] The ontouml-vp-plugin is available at https://github.com/OntoUML/ontouml-vp-plugin.

[12] The Visual Paradigm UML CASE tool is a commercial tool also available at the moment through community licenses for non-commercial purposes.

[13] The ontouml-schema is available at https://github.com/OntoUML/ontouml-sche,a.

[14] The ontouml-js is available at https://github.com/OntoUML/ontouml-js.

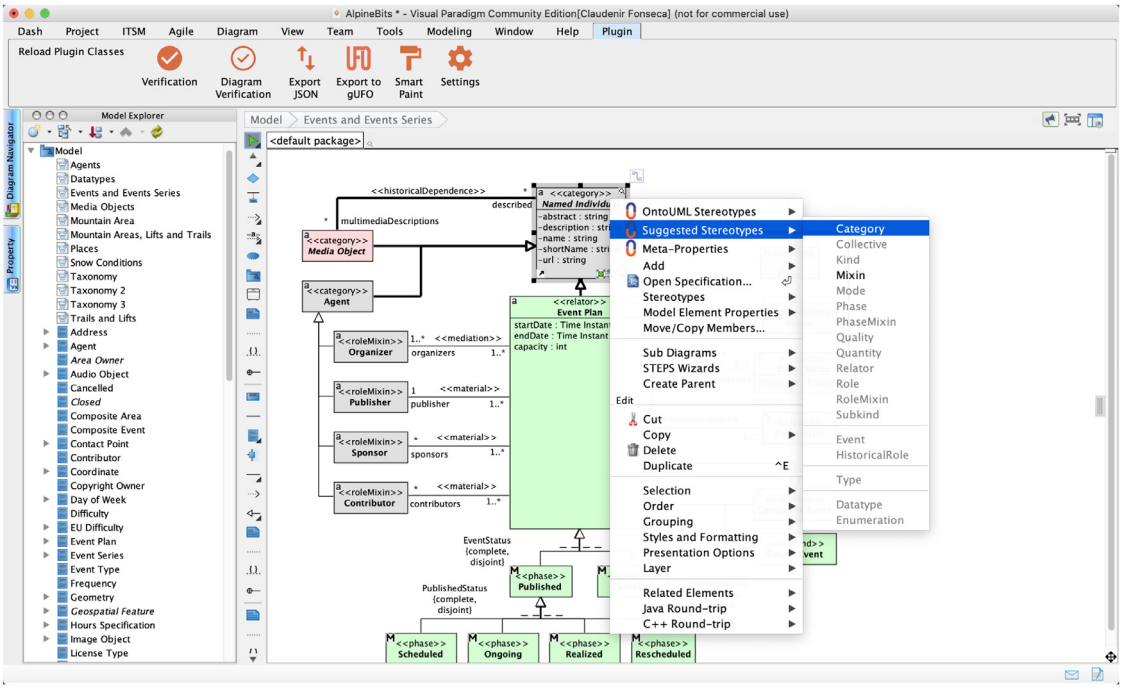


Fig. 16. Screenshot of Visual Paradigm's interface with the ontouml-vp-plugin installed.

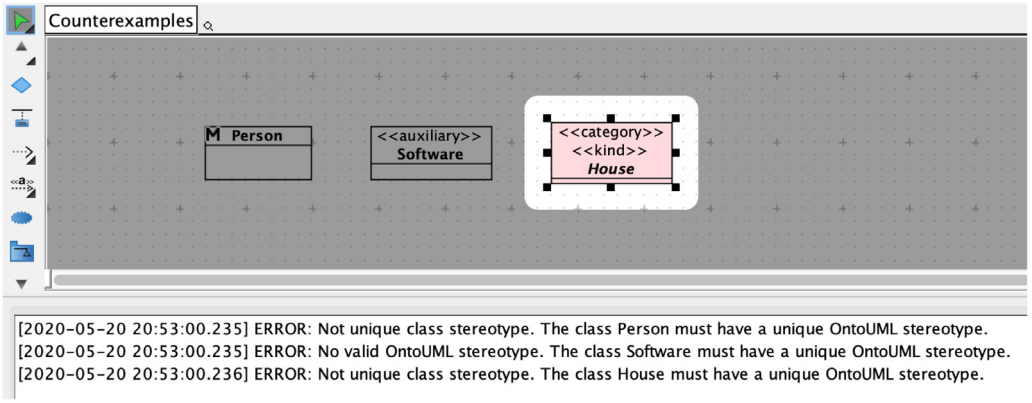


Fig. 17. Screenshot of Visual Paradigm where the ontouml-vp-plugin presents to the modeler some verification issues.

for categories, individuals and their properties without the meta-linguistic support available to second language learners lies on the mapping between the language-specific symbols for these categories onto a system of categories already existing in the language of thought, i.e., a *cognitive ontology*.

In this section, we discuss some empirical evidence supporting many of the points defended throughout this paper. This evidence results from a number of psychological experiments in the area of cognitive psychology, which has been developed with the aim of investigating categorical development and logical competence in infants since the pre-language age of 3–4 months. Firstly, many laboratory results provide evidence that infants in the early age of 3–4 months are already able to form categories (e.g., [42]). Cohen and Younger [43], and Macnamara [44], for instance, provide evidences that children are able to classify objects into categories for which they have no natural-language symbols. The reason that categorization appears so early in cognitive development is related to its fundamental relevance to cognition. As [45] puts it: *"Categorization [...] is a means of simplifying the environment, of reducing the load on memory, and of helping us to store and retrieve information efficiently"*. Without concepts, mental life would be chaotic. If we perceived each entity as unique, we would be overwhelmed by the sheer diversity of what we experience and unable to remember more than a minute fraction of what we encounter. Furthermore, if each individual entity needed a distinct name, our language would be staggeringly complex and communication virtually impossible. In contrast, if you know nothing about a novel object but you are told it is an instance of X, you can infer that the object has all or many properties that Xs have [46].

Nonetheless, in order to enable the learning of what are the properties that we can expect instances of X to have, another ability, namely the ability to (re)identity instances of X must be present. If all of an object's properties were immediately manifest to the child upon every encounter, there would be no need to learn and remember what these properties were. However, carrying the knowledge of substances becomes necessary since most of a substance's properties are not manifest but hidden from us most of the time [47]. For example, different encounters with Felix will reveal different properties about the individual Felix and about the kind Cat. However, for this to happen, the child must be able to:

(i) recognize that Felix *is a Cat*;
(ii) recognize that the individual that can jump from the table to the TV set *is the same* as the one that can be fed milk;
(iii) recognize that Tom *is also a Cat*, thus, he can also be fed milk.

In summary, both principles of application and identity are fundamental in our construal of the environment.

A number of experiments provide evidence for the existence of an individuation and identity system in infants by the age they begin to form categories. Researchers such as [48–50] provide evidence that by the early age 3–4 months old infants have criteria for deciding whether an object is the same one as a previously seen object. For instance, [48] provides evidence that until about nine months of age, infants rely on a unique principle of individuation and identity for all objects, which is supplied by the type *Physical Object*. This notion of physical object is synonymous to *maximally connected physical object whose parts move along together in a spatiotemporal continuous path*. As defended by the authors, in this sense, physical object is indeed a sortal since any identity and individuation statements involving two physical objects is determinate. They term this system of individuation an object-based system. These results show that not only does spatiotemporal discontinuity lead to a representation of two distinct objects, but also that spatiotemporal continuity leads to a representation of a single, persisting object. Other laboratories have also replicated this basic finding using somewhat different procedures (e.g., [51]).

Thus, even young infants have some criteria for establishing representations of distinct objects. These first criteria are spatiotemporal in nature, including generalizations such as:

(i) objects travel on spatiotemporally connected paths;
(ii) two objects cannot occupy the same space at the same time, and
(iii) one object cannot be in two places at the same time [36].

Xu and Baker [52] address the question of whether infants can use perceptual property information for object individuation, i.e., if non-sortal categories can support the judgment of individuation statements. The results show that, for 10-month infants, perceptual properties are at best only used to confirm the application of the principle of identity which is first supplied by the sortal *Physical Object*. Moreover, these results show that *spatiotemporal evidence for a single object changing properties overrides perceptual property information* (see also [48]). The work of [36] also shows that 12-month infants are not able to use non-sortal categories alone to support the judgment of individuation statements. This view is also supported by [47] who calls the attention for the fact that children come to appreciate separable dimensions, such as color, shape, and size only after a considerable period in which "holistic similarities" dominate their attention.

Results from [36] show that, between 9 to 12 months of age, a second system of individuation emerges in infant's cognition. This is named a *kind-based individuation system* and operates independently of the object-based attention system. In general, by the early age of 12 months, infants have already developed the multiple-sortal system which is used by adults to judge individuation and identity statements. Moreover, [53] shows that:

(i) representations of object kinds can override strong spatiotemporal evidence for a single object;
(ii) perceptual property information is always treated as kind-relative.

As remarked by the author, *"during this period, infant's worldview undergoes fundamental changes: They begin with a world populated with objects [...]. By the end of the first year of life, they begin to conceptualize a world populated with sortal-kinds [...]. In this new world, objects are thought of not as 'qua object' but rather 'qua dog' or 'qua table'."* Other experiments such as those conducted in [54,55], and [56] corroborate with these findings and support the claim that around 12-months infants are sensitive to the distinction between sortals and arbitrary general terms, which are represented differently and used differently in individuation tasks.

The work in [53] also suggests that it is not a coincidence that along with acquiring their first words, infants begin to develop their kind-based system of individuation. The bulk of a child's first words are concrete nouns, including proper names and names for sortal universals [47]. For example, [57] shows that children learn common nouns (the linguistic counterparts of sortal universals) before they learn predicates such as verbs and adjectives (counterparts of characterizing universals). [58] presents evidence that this finding holds cross-culturally for children learning German, Kaluli, Japanese, Mandarin Chinese, Turkish, and English. In summary, in the early stages of language learning, children are more likely to pick words for sortals than of other kinds. Although perceived, attributes and events are construed, individuated, and conceptualized under the dependence of a sortal [35].

[59] and [44] provide evidence that children younger than 17-month-old are able to distinguish proper names by coordinating the notions of individual and kind. According to these findings, children are able to judge that individuals of some kinds, but not of others, are likely to be the bearers of a proper name. Together with the results from [60], these findings provide strong indications that under certain circumstances children are led to take some words as proper names (namely, when applied to individuals of familiar kinds) and in others to take them as sortals (when applied to individuals of unfamiliar kinds). [35] advocates that this

evidence is an obvious suggestion that children have the appropriate sortals to support the learning of proper names in some language. In addition to that, when reporting on a number of observations extracted from the linguistic record he kept of his son, Macnamara remarks that his son's use of proper names behaved like rigid designators from the start [35].

According to [61,62], when learning a word for a kind of object, basic-level sortals (substance sortals) are the ones that occur to children. For example, when creating categories, children attend to similarities among dogs before subclassifying them and before they attend to properties dogs share with other animals. [35] strongly argues that substance sortals hold a psychologically salient and privileged position compared to other types of rigid sortals, and that children's perceptual systems seem to be especially tuned to identify substance sortals. This position finds evidence in the results of [63,64].

In summary, a substantial number of psychological experiments confirm the philosophy of language thesis that individuation and identity judgment can only take place with the support of a sortal universal. Both systems of individuation and identity that are employed by human cognition are sortal-based. Humans start with a principle of identity afforded by the unique sortal physical object and, in a later developmental phase, undergo a cognitive shift to a system that employs a multitude of principles of identity supplied by different substance sortals. In both systems, perceptual property information is secondary and can be overridden. [53] defends the position that this developmental process makes good sense in terms of learnability, since it starts the child on the solid ground of a concept of object that seems to be innate [49] and it allows the child to work with these individuated objects and with the help of language, ultimately develop a new ontology of sortal-kinds. In addition, as defended by [47], the primacy of substance sortals also makes good sense in evolutionary terms since, from the standpoint of an organism that wishes to learn, the focus should be on constructing categorizations that are essential, since they are the ones that supply knowledge that affords the most useful and reliable inferences.

Finally, we should highlight that most of the aforementioned empirical studies focus on substantials as opposed to other types of endurants. This is understandable, given that these are experiments conducted with children (often pre-language acquisition) and, hence, reliant on concrete physical objects (e.g., a rubber duck, a red ball), which are examples of substantials. Endurants such as qualities, modes, and relators are proper called *abstract particulars* in the literature as they are ultimately abstractions over properties of substantials, and, thus, *identificationally dependent* on the latter (e.g., in order to identify the concrete redness of that particular ball, we have to first identity the ball) [3]. Nonetheless, as pointed out by [65], there is solid evidence for these abstract particulars in the literature. On one hand, in the analysis of the content of perception [66,67], abstract particulars such as colors, sounds, runs, laughter, and singings are the immediate objects of everyday perception. On the other hand, the idea of having these entities as *truthmakers* [20,67,68] underlies a standard event-based approach to natural language semantics, as initiated by Davidson [69] and Parsons [70]. In particular, there is strong support from linguistic data for the presence in our cognitive ontology of existentially dependent endurants and, in particular, ones that can qualitatively change in time while maintaining their identity. [68,71,72]. For example, Moltmann [68] uses the notion of *variable tropes* (existentially dependent particulars that can change while maintaining identity) to address a number of fundamental phenomena in cognition and language. Once we have that:

(i) existentially dependent particularized properties are countenanced by our cognitive ontology (or *Natural Language Ontology* to use Moltmann's term [73]);
(ii) these are genuine endurants that can change and can be the bearer of modal properties, i.e., that can bear essential and accidental property.

Then, it just follows logically that: non-rigid types also apply to them; taxonomies involving their types can be organized (refactored) in terms of non-sortal properties (i.e., rigid and non-rigid properties that cross the boundaries of kinds).

8. Final considerations

In this paper, we contribute to the ontological foundations of conceptual modeling by proposing a formal theory of endurant types and the taxonomic structures involving them. This theory was developed to address several empirically elicited requirements, collected from observing the practice of the OntoUML community, while using these notions to model a variety of domains (*claim to relevance*). Despite the empirical origin of these requirements, they are very much in line with the literature in philosophy, linguistics, and cognitive science (*claim to theoretical adequacy*). Additionally, this formal theory has been checked for its consistency using a theorem prover (*claim to consistency*).

The theory proposed here addresses an important fragment of a new version of the Unified Foundational Ontology (UFO) and serves as a basis for a new version of the ontology-driven conceptual modeling language OntoUML. In particular, the precise relation between language and ontology is exercised here via a design process that uses: the ontological distinctions put forth by the theory to derive the metamodel of the language; the formal axioms and ontological constraints of the theory to derive semantic and, ultimately, syntactical constraints for this language. This process is at the core of classical approaches on ontology-based language engineering [3]. As a result of applying this process, we manage to use this theory to construct two artifacts (*claim to realizability*), namely: a new UML profile (a lightweight extension to the UML 2.0 metamodel) capturing the concepts and formal constraints proposed by this theory; and a computational tool implementing this profile. In order to demonstrate the wider applicability of this approach, we also employed the same method to design a third artifact, namely: an infrastructural specification implemented in OWL 2 DL that represents the categories put forth by our theory; a set of SPARQL constraints enriching that specification, and that allows for an automated verification service checking the compliance of Semantic Web ontologies (built according to this specification) to the constraints proposed by our theory (*claim to realizability and non-specificity*).

The work developed here focuses exclusively on endurant types and taxonomic relations. In extensions of this work, we addressed other relations involving endurants and endurant types (e.g., relational dependence for roles and role mixins, existential dependence for moment types, the foundation between relators and events [74], as well as the relation between endurant types and event types [75]). More broadly, the work presented is part of a research program aimed at addressing a fuller evolution of UFO and OntoUML, which ultimately aims at providing foundations and engineering tools for advancing the theory and practice of conceptual modeling.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brazil (CAPES) – Finance Code 001 (grant 23038.028816/2016-41). João Paulo A. Almeida is also supported by CNPq grants 312123/2017-5 and 407235/2017-5. Claudenir M. Fonseca, Giancarlo Guizzardi, and Tiago Prince Sales are supported by the NeXON Project (Free University of Bozen-Bolzano). Alessander Botti Benevides was supported by FAPES postdoctoral grant Nº 83740910/18.

References

[1] M. Bunge, Treatise on Basic Philosophy. Ontology I: The Furniture of the World, Riedel, Boston, 1977.
[2] J.J.M. Leeuwen, Individuals and Sortal Concepts: an Essay in Logical Descriptive Metaphysics Ph.D. thesis, University of Amsterdam, 1991.
[3] G. Guizzardi, Ontological Foundations for Structural Conceptual Models (Ph.D. thesis), Telematica Instituut / CTIT, 2005.
[4] R. Wieringa, W. de Jonge, P. Spruit, Using dynamic classes and role classes to model object migration, Theory Pract. Object Syst. 1 (1) (1995) 61–83.
[5] G. Booch, R.A. Maksimchuk, M.W. Engle, B.J. Young, J. Connallen, K.A. Houston, Object-oriented analysis and design with applications, ACM SIGSOFT Softw. Eng. Notes 33 (5) (2008) 29.
[6] C. Bock, J. Odell, A more complete model of relations and their implementation: Roles, J. Object-Oriented Program. 11 (2) (1998) 51–54.
[7] F. Steimann, On the representation of roles in object-oriented and conceptual modelling, Data Knowl. Eng. 35 (1) (2000) 83–106.
[8] J. Evermann, Y. Wand, Towards ontologically based semantics for uml constructs, in: International Conference on Conceptual Modeling, Springer, 2001, pp. 354–367.
[9] N. Guarino, A concise presentation of ITL, SIGART Bull. 2 (3) (1991) 61–69, http://dx.doi.org/10.1145/122296.122305.
[10] C.A. Welty, N. Guarino, Supporting ontological analysis of taxonomic relationships, Data Knowl. Eng. 39 (1) (2001) 51–74, http://dx.doi.org/10.1016/S0169-023X(01)00030-1.
[11] G. Guizzardi, G. Wagner, N. Guarino, M. van Sinderen, An ontologically well-founded profile for UML conceptual models, in: Advanced Information Systems Engineering, 16th International Conference, CAiSE 2004, Riga, Latvia, June 7-11, 2004, Proceedings, 2004, pp. 112–126, http://dx.doi.org/10.1007/978-3-540-25975-6_10.
[12] A. Gupta, The Logic of Common Nouns: An Investigation in Quantified Modal Logic, Yale University Press, 1980.
[13] J. Macnamara, G.E. Reyes, et al., The Logical Foundations of Cognition, Vol. 4, Oxford University Press on Demand, 1994.
[14] G. Guizzardi, H. Herre, G. Wagner, On the general ontological foundations of conceptual modeling, in: S. Spaccapietra, S.T. March, Y. Kambayashi (Eds.), Conceptual Modeling — ER 2002, Springer, Berlin, Heidelberg, 2003, pp. 65–78.
[15] G. Guizzardi, G. Wagner, J.P.A. Almeida, R.S. Guizzardi, Towards ontological foundations for conceptual modeling: the unified foundational ontology (UFO) story, Appl. Ontol. 10 (3–4) (2015) 259–271, http://dx.doi.org/10.3233/AO-150157.
[16] M. Verdonck, F. Gailly, Insights on the use and application of ontology and conceptual modeling languages in ontology-driven conceptual modeling, in: I. Comyn-Wattiau, K. Tanaka, I.-Y. Song, S. Yamamoto, M. Saeki (Eds.), Conceptual Modeling, Springer, Cham, 2016, pp. 83–97, http://dx.doi.org/10.1007/978-3-319-46397-1_7.
[17] M. Verdonck, Ontology-Driven Conceptual Modeling: Model Comprehension, Ontology Selection, and Method Complexity (Ph.D. thesis), Applied Economics Department, Ghent University, Belgium, 2018.
[18] I. Blums, H. Weigand, Financial Reporting by a Shared Ledger, in: Proceedings of the Joint Ontology Workshops 2017 Episode 3: The Tyrolean Autumn of Ontology, Vol. 2050, Bozen-Bolzano, Italy, September 21-23, 2017, 2017.
[19] A.P. do Carmo, T. Zamperini, M.R. de Mello, A.L. de Castro Leal, A.S. Garcia, Ontologia das coisas para espaços inteligentes baseados em visão computacional, in: 9th Seminar on Ontology Research in Brazil, 2016, pp. 181–186.
[20] N. Guarino, G. Guizzardi, "We need to discuss the relationship": Revisiting relationships as modeling constructs, in: J. Zdravkovic, M. Kirikova, P. Johannesson (Eds.), Advanced Information Systems Engineering, Springer International Publishing, Cham, 2015, pp. 279–294, http://dx.doi.org/10.1007/978-3-319-19069-3_18.
[21] V. Zamborlini, A. Betti, C. van den Heuvel, Toward a core conceptual model for (im)material cultural heritage in the golden agents project, in: A. Fensel, L. Daniele (Eds.), Joint Proceedings of SEMANTiCS 2017 Workshops, Vol. 2063, 2017.
[22] OMG Unified Modeling Language (OMG UML) Version 2.5.1, Object Management Group (OMG), 2017, https://www.omg.org/spec.
[23] G. Guizzardi, C.M. Fonseca, A.B. Benevides, J.P.A. Almeida, D. Porello, T.P. Sales, Endurant types in ontology-driven conceptual modeling: Towards ontouml 2.0, in: J.C. Trujillo, K.C. Davis, X. Du, Z. Li, T.W. Ling, G. Li, M.L. Lee (Eds.), Conceptual Modeling. ER 2018, Springer, Cham, 2018, pp. 136–150.
[24] Object Constraint Language: Version 2.4, Object Management Group (OMG), 2014.
[25] E.J. Lowe, More Kinds of Being: A Further Study of Individuation, Identity, and the Logic of Sortal Terms, John Wiley & Sons, 2015.
[26] E. Hirsch, The Concept of Identity, Oxford University Press, 1992.
[27] Z. Rybola, Towards OntoUML for Software Engineering: Transformation of OntoUML into Relational Databases, Czech Technical University in Prague, 2017.
[28] V.A. Carvalho, J.P.A. Almeida, C.M. Fonseca, G. Guizzardi, Multi-level ontology-based conceptual modeling, Data Knowl. Eng. 109 (2017) 3–24, http://dx.doi.org/10.1016/j.datak.2017.03.002.
[29] M. Fitting, R.L. Mendelsohn, First-Order Modal Logic, Vol. 277, Springer Science & Business Media, 2012.
[30] W.V.O. Quine, et al., Ontological relativity, Ontol. Relat. Other Essays 15 (1969) 26–68.
[31] D. Wiggins, et al., Sameness and Substance Renewed, Cambridge University Press, 2001.

[32] V.A. Carvalho, J.P.A. Almeida, G. Guizzardi, Using reference domain ontologies to define the real-world semantics of domain-specific languages, in: M. Jarke, J. Mylopoulos, C. Quix, C. Rolland, Y. Manolopoulos, H. Mouratidis, J. Horkoff (Eds.), Advanced Information Systems Engineering, Springer, Cham, 2014, pp. 488–502, http://dx.doi.org/10.1007/978-3-319-07881-6_33.
[33] A. Lanusse, Y. Tanguy, H. Espinoza, C. Mraidha, S. Gerard, P. Tessier, R. Schnekenburger, H. Dubois, F. Terrier, Papyrus UML: an open source toolset for MDA, in: Proc. of the Fifth European Conference on Model-Driven Architecture Foundations and Applications (ECMDA-FA 2009), 2009, pp. 1–4.
[34] OWL 2 Web Ontology Language Document Overview: W3C Recommendation 27 October 2009, OWL Working Group and others, 2009.
[35] J.T. Macnamara, A Border Dispute the Place of Logic in Psychology, The MIT Press, 1986.
[36] F. Xu, S. Carey, N. Quint, The emergence of kind-based object individuation in infancy, Cogn. Psychol. 49 (2) (2004) 155–190.
[37] J. Lidz, S. Waxman, J. Freedman, What infants know about syntax but couldn't have learned: experimental evidence for syntactic structure at 18 months, Cognition 89 (3) (2003) 295–303.
[38] N. Chomsky, Aspects of the Theory of Syntax, Vol. 11, MIT press, 2014.
[39] N. Chomsky, Rules and representations, Behav. Brain Sci. 3 (1) (1980) 1–15.
[40] N. Chomsky, Knowledge of Language: Its Nature, Origin, and Use, Greenwood Publishing Group, 1986.
[41] J.A. Fodor, The Language of Thought, Vol. 5, Harvard university press, 1975.
[42] P.D. Eimas, P.C. Quinn, Studies on the formation of perceptually based basic-level categories in young infants, Child Dev. 65 (3) (1994) 903–917.
[43] L.B. Cohen, B.A. Younger, Perceptual categorization in the infant, New Trends Concept. Represent. (1983).
[44] J. Macnamara, Names for Things: A Study of Human Learning, Mit Press, 1982.
[45] E.M. Markman, Categorization and Naming in Children: Problems of Induction, Mit Press, 1989.
[46] E.E. Smith, D.L. Medin, Categories and Concepts, Vol. 9, Harvard University Press Cambridge, MA, 1981.
[47] R.G. Millikan, A common structure for concepts of individuals, stuffs, and real kinds: More mama, more milk, and more mouse, Behav. Brain Sci. 21 (1) (1998) 55–65.
[48] F. Xu, S. Carey, Infants' metaphysics: The case of numerical identity, Cogn. Psychol. 30 (2) (1996) 111–153.
[49] E.S. Spelke, R. Kestenbaum, D.J. Simons, D. Wein, Spatiotemporal continuity, smoothness of motion and object identity in infancy, Br. J. Dev. Psychol. 13 (2) (1995) 113–142.
[50] M.K. Moore, R. Borton, B.L. Darby, Visual tracking in young infants: Evidence for object identity or object permanence?, J. Exp. Child Psychol. 25 (2) (1978) 183–198.
[51] A. Aguiar, R. Baillargeon, 2.5-month-old infants' reasoning about when objects should and should not be occluded, Cogn. Psychol. 39 (2) (1999) 116–157.
[52] F. Xu, A. Baker, Object individuation in 10-month-old infants using a simplified manual search method, J. Cogn. Dev. 6 (3) (2005) 307–323.
[53] F. Xu, Categories, kinds, and object individuation in infancy, in: Building Object Categories in Developmental Time, Psychology Press, 2005, pp. 81–108.
[54] L. Bonatti, E. Frot, R. Zangl, J. Mehler, The human first hypothesis: Identification of conspecifics and individuation of objects in the young infant, Cogn. Psychol. 44 (4) (2002) 388–426.
[55] S.R. Waxman, D.B. Markow, Words as invitations to form categories: Evidence from 12-to 13-month-old infants, Cogn. Psychol. 29 (1995) 257–302.
[56] A.E. Booth, S.R. Waxman, Mapping words to the world in infancy: Infants' expectations for count nouns and adjectives, J. Cogn. Dev. 4 (3) (2003) 357–381.
[57] J. Macnamara, Cognitive basis of language learning in infants, Psychol. Rev. 79 (1) (1972) 1.
[58] D. Gentner, Why Nouns are Learned Before Verbs: Linguistic Relativity Versus Natural Partitioning, Center for the Study of Reading Technical Report; no. 257, 1982.
[59] N. Katz, E. Baker, J. Macnamara, What's in a name? A study of how children learn common and proper names, Child Dev. (1974) 469–473.
[60] S.A. Gelman, M. Taylor, How two-year-old children interpret proper and common names for unfamiliar objects, Child Dev. (1984) 1535–1540.
[61] E. Rosch, C.B. Mervis, W.D. Gray, D.M. Johnson, P. Boyes-Braem, Basic objects in natural categories, Cogn. Psychol. 8 (3) (1976) 382–439.
[62] E.M. Markman, J.E. Hutchinson, Children's sensitivity to constraints on word meaning: Taxonomic versus thematic relations, Cogn. Psychol. 16 (1) (1984) 1–27.
[63] J.M. Anglin, Word, Object, and Conceptual Development, WW Norton, 1977.
[64] C.B. Mervis, M.A. Crisafi, Order of acquisition of subordinate-, basic-, and superordinate-level categories, Child Dev. (1982) 258–266.
[65] L. Schneider, Formalised Elementary Formal Ontology, Tech. Rep., ISIB-CNR, 2002.
[66] E.J. Lowe, The Possibility of Metaphysics: Substance, Identity, and Time, Clarendon Press, 1998.
[67] K. Mulligan, P. Simons, B. Smith, Truth-makers, Philos. Phenomenol. Res. 44 (3) (1984) 287–321.
[68] F. Moltmann, Events, tropes, and truthmaking, Philos. Stud. 134 (3) (2007) 363–403, http://dx.doi.org/10.1007/s11098-005-0898-4.
[69] D. Davidson, The logical form of action sentences, in: N. Rescher (Ed.), The Logic of Decision and Action, University of Pittsburgh Press, 1967.
[70] J. Parsons, Distributional properties, Lewisian Themes (2004) 173–180.
[71] F. Moltmann, Intensional verbs and their intentional objects, Nat. Lang. Semant. 16 (3) (2008) 239–270.
[72] F. Moltmann, Degree structure as trope structure: A trope-based analysis of positive and comparative adjectives, Linguist. Philos. 32 (1) (2009) 51–94.
[73] F. Moltmann, Natural language ontology, in: Oxford Encyclopedia of Linguistics, Oxford University Press, 2017, http://dx.doi.org/10.1093/acrefore/9780199384655.013.330.
[74] C.M. Fonseca, D. Porello, G. Guizzardi, J.P.A. Almeida, N. Guarino, Relations in ontology-driven conceptual modeling, in: International Conference on Conceptual Modeling, Springer, 2019, pp. 28–42.
[75] J.P.A. Almeida, R.A. Falbo, G. Guizzardi, Events as entities in ontology-driven conceptual modeling, in: International Conference on Conceptual Modeling, Springer, 2019, pp. 469–483.

Giancarlo Guizzardi is a Professor of Computer Science at the Free University of Bozen-Bolzano, Italy, where he leads the Conceptual and Cognitive Modeling Research Group (CORE). He is also a Professor of Software Science and Evolution at the University of Twente, The Netherlands. He has been active for more than two decades in the areas of Ontologies, Conceptual Modeling, and Information Systems Engineering. He is currently an associate editor for the Applied Ontology journal and for Data & Knowledge Engineering and a member of a number of international journal editorial boards. He is a member of the ER Steering Committee and of the Advisory Board of the International Association for Ontology and its Applications (IAOA).

Claudenir M. Fonsica is a Ph.D. student at the Free University of Bolzano and member of the Conceptual and Cognitive Modelling Research Group (CORE). He holds a MSc. Degree in Computer Sciences at the Federal University of Espírito Santo, where he developed research as member of the Ontology and Conceptual Modeling Research Group (NEMO). Claudenir has been working in the latest years on building, formalizing, and validating core ontologies, as well as developing software support for ontology-driven conceptual modeling. He is currently interested in ontology-driven conceptual modeling, multi-level modeling, and the application of domain-specific languages for ontology specification.

João Paulo A. Almeida is Associate Professor at the Federal University of Espírito Santo, Brazil, and Senior Member of the Ontology and Conceptual Modeling Research Group (NEMO). He holds a Ph.D. in Computer Science from the University of Twente, The Netherlands. For over a decade, he has been working on

the application of ontologies in conceptual modeling, enterprise architecture and enterprise modeling. Particularly of interest to this project, he has a decade long experience in the development of Model-Based Engineering tools and was directly involved in the development of a number of computational tools for OntoUML. He is a senior member of the IEEE and the ACM. He serves as a member of the IEEE EDOC Conference Steering Committee and of the Advisory Board of the International Association for Ontology and its Applications (IAOA).

Tiago Prince Sales is an Assistant Professor at the Conceptual and Cognitive Modelling Research Group (CORE) of the Faculty of Computer Science, Free University of Bozen-Bolzano, Italy. He received his PhD from the University of Trento, Italy, with a cum laude distinction and the additional qualification of doctor europaeus. His main research focus is on ontology-driven conceptual modeling. In particular, he works on the development and evolution of the OntoUML language, he investigates and develops engineering tools for conceptual modeling, such as model simulation and anti-patterns, and he develops well-founded reference ontologies for business and social domains, such as value, risk, competition, and trust. He has over ten years of experience in industrial and technology transfer projects in a wide range of domains, such as tourism, media asset management, public healthcare, transportation, and oil and gas.

Alessander Botti Benevides is an Infrastructure Analyst, a public servant at the Vila Velha Town Hall (PMVV), Brazil. Prior to that, from 2015 to 2020, he was a postdoctoral researcher at the Graduate Program in Computer Science (PPGI) of the Federal University of Espírito Santo (UFES), Brazil. He holds a Ph.D. from the International Doctoral School on Information and Communication Technology of the University of Trento (UNITN), Italy. During his Ph.D. studies, he spent three months abroad at the University of Exeter, England (8/2014 - 10/2014). He graduated in Computer Engineering in 2007 and received his master's degree in Informatics in 2010, both by UFES. He has experience in ontological analysis; Formal Ontology; formalization in first-order logic, modal logics, and description logics (in particular, $\mathcal{SROIQ}$, the formal underpinning of OWL2); analysis and specification of structural and behavioral constraints by means of the declarative language Alloy; using SAT solvers; programming in Haskell; and simulating logical theories by means of the automatic generation of models, having built a graphical editor for the building, verification and validation of OntoUML models.

Dr. Daniele Porello is a Researcher at the Laboratory for Applied Ontology, Institute for Cognitive Sciences and Technologies, Italian National Research Council (CNR). He received the Ph.D. from the University of Genova. He worked at the Institute National de Recherche ÌĄen Informatique et en Automatique (INRIA) in Bordeaux, at the Institute for Logic, Language and Computation (ILLC) in Amsterdam, and at the KRDB Research Centre of the Free University of Bozen-Bolzano. He has been working in areas such as logics for multiagent systems, computational social choice, foundational and applied ontology, computational logics, non-classical logics, and cognitive semantics.