

# Endurant Types in Ontology-Driven Conceptual Modeling: Towards OntoUML 2.0

Giancarlo Guizzardi<sup>1,2</sup>, Claudenir M. Fonseca<sup>1</sup>, Alessandro Botti Benevides<sup>2</sup>,  
João Paulo A. Almeida<sup>2</sup>, Daniele Porello<sup>1</sup>, and Tiago Prince Sales<sup>3</sup>

<sup>1</sup> Conceptual and Cognitive Modeling Research Group (CORE), Free University of  
Bozen-Bolzano, Italy

`{giancarlo.guizzardi,daniele.porello,cmoraisfonseca}@unibz.it`

<sup>2</sup> NEMO, Federal University of Espírito Santo, Brazil

`abbenevides@inf.ufes.br`

<sup>3</sup> DISI, University of Trento, Italy

`tiago.princesales@unitn.it`

**Abstract.** 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. This ontology has led to the development of an Ontology-Driven Conceptual Modeling language dubbed OntoUML, reflecting the ontological micro-theories comprising UFO. Over the years, UFO and OntoUML have been successfully employed in a number of academic, industrial and governmental settings to create conceptual models in a variety of different domains. These experiences have pointed out to opportunities of improvement not only to the language itself but also to its underlying theory. In this paper, we take the first step in that direction by revising the theory of types in UFO in response to empirical evidence. The new version of this theory shows that many of the meta-types present in OntoUML (differentiating Kinds, Roles, Phases, Mixins, etc.) 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 a formal characterization of this fragment of the theory, which is then used to advance a metamodel for OntoUML 2.0. Finally, we propose a computational support tool implementing this updated metamodel.

**Keywords:** OntoUML, UFO, Ontology-Driven Conceptual Modeling

## 1 Introduction

In recent years, there has been a growing interest in the use of foundational ontologies (i.e., ontological theories in the philosophical sense) to evaluate and (re)design conceptual modeling languages. For over a decade now, a community of researchers has contributed to the development of a foundational ontology (termed the Unified Foundational Ontology – UFO) aimed at providing foundations underlying all conceptual modeling major constructs. This ontology has

also been systematically used to design an ontology-driven conceptual modeling (ODCM) language termed OntoUML [9,8]. UFO and OntoUML have been successfully employed in academic, industrial and governmental settings to create conceptual models in a number 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 [9]. In fact, research shows that they are among the most used foundational ontology and modeling language in the ODCM literature, respectively [13]. Moreover, empirical evidence shows that OntoUML significantly contributes to improving the quality of conceptual models without requiring an additional effort to produce them. For instance, the work of [12] 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).

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<sup>1</sup>. In particular, we have managed to observe a number of different ways in which people would slightly subvert the syntax of the language, ultimately creating what we could call “*systematic subversions*” of the language [9]. These “subversions” would (purposefully) produce models that were grammatically 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” led us in this article to reconsider some of the theoretical foundations underlying the language, i.e., it led us to rethink and evolve a core theory in UFO, namely, its theory of *Endurant Types and Taxonomic structures* [8].

Structural conceptual modeling languages (e.g., ER, UML, ORM but also OntoUML) are designed to model types whose instances are endurants (object-like entities). These can be of the regular independent sort but also *dependent* endurants (e.g., weak entities/objects and objectified relationships). In ODCM approaches, such as OntoUML (but also OntoClean [7]), there are finer-grained distinctions among types, for instance, differentiating endurant types according to meta-types such as *Kinds*, *Phases*, *Roles* and *Mixins*.

Following the original version of UFO’s theory of types, many of these meta-types present in OntoUML are currently restricted to the modeling of Substantial types (independent objects). For instance, we can represent that “Person” is the *Kind* of entity Mick Jagger is, 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. But, now, how can we model that the relationship between

---

<sup>1</sup> Several dozens of these models are available at <http://www.menthor.net/>

Giovanni and 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)[1,2,6,14]. 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.

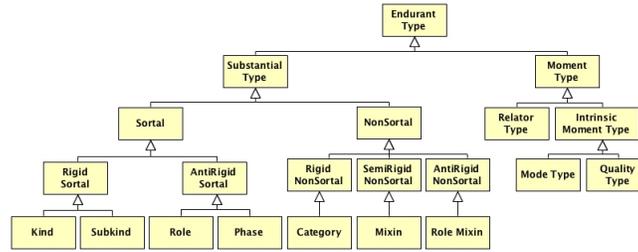
The contributions of this paper are three-fold. First, we propose a new formal theory of endurant types and taxonomic structures for UFO. Although developed in the framework of UFO, this theory amounts to a contribution to ODCM, more broadly. In particular, we can influence approaches such as OntoClean and ORM, 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 [8], we employ this new formalized version of the theory to advance an enhanced metamodel for OntoUML 2.0. Finally, we employ this metamodel to implement a software tool for OntoUML 2.0 supporting model verification.

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 theory, UFO; Section 4 introduces a new version of OntoUML, presenting its constructs and syntactic constraints; Section 5 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 [8], respectively. Substantials are existentially independent objects such as John Lennon, the Moon, an organization, a car, a dog. 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 con-



**Fig. 1.** A Taxonomy of Endurant Types in UFO.

necting 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 [8].

Concerning the substantial type hierarchy, sortal types are the ones that either provide or carry a uniform *principle of identity* for their instances. A principle of identity regarding a sortal  $S$  makes explicit the properties that no two instances of  $S$  can have in common, because 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 [8]. 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 are sortal rigid types that provide a uniform principle of identity for their instances (e.g., Person). 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 [8]. 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* [8].

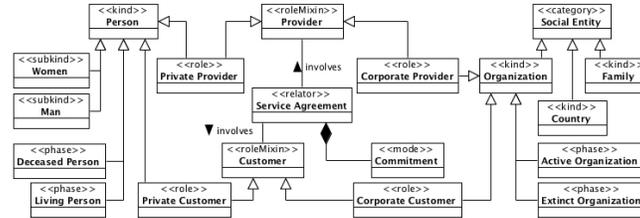


Fig. 2. OntoUML example.

Non-Sortals (also called *dispersive types* [8]) 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; 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, legislations). 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 [8] (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)[8]. Although not prescribed in the original set of UFO enduring types, over the years, the notion of a Phase Mixin emerged as a useful notion that was found missing by different authors [11,4]. 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 [8]. An example of a model illustrating these notions is presented in Fig. 2.

### 3 A New Formal Theory of Endurant Types

In this section, we present a first-order modal theory of enduring 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 plus the Barcan formula and its converse suffices [5]. That means that we assume a *fixed* domain of entities for every possible world, what is traditionally associated to a *possibilistic* view of the entities of the domain, i.e., the domain includes all the *possibilia*. In the following formulas, we drop both the universal quantifier and the 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 meaning. This theory has been specified and verified in TPTP (<http://www.tptp.org>), being automatically proven with provers SPASS 3.9 and Z3 4.4.1.<sup>2</sup>

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).

$$\begin{array}{ll}
\mathbf{a1} & \text{Type}(x) \leftrightarrow \Diamond(\exists y(y :: x)) \\
\mathbf{a2} & \text{Individual}(x) \leftrightarrow \Box(\neg \exists y(y :: x)) \\
\mathbf{t2} & \neg \exists x(\text{Individual}(x) \wedge \text{Type}(x)) \\
\mathbf{a4} & x \sqsubseteq y \leftrightarrow \text{Type}(x) \wedge \text{Type}(y) \wedge \Box(\forall z(z :: x \rightarrow z :: y)) \\
\mathbf{t3} & x \sqsubseteq y \rightarrow (x \sqsubseteq x \wedge y \sqsubseteq y) \\
\mathbf{a5} & \forall t_1, t_2, x((x :: t_1 \wedge x :: t_2 \wedge \neg(t_1 \sqsubseteq t_2) \wedge \neg(t_2 \sqsubseteq t_1)) \rightarrow (\exists t_3(t_1 \sqsubseteq t_3 \wedge t_2 \sqsubseteq t_3 \wedge x :: t_3) \vee \exists t_3(t_3 \sqsubseteq t_1 \wedge t_3 \sqsubseteq t_2 \wedge x :: t_3))) \\
\mathbf{a3} & x :: y \rightarrow \text{Individual}(x) \wedge \text{Type}(y) \\
\mathbf{t1} & \text{Individual}(x) \vee \text{Type}(x) \\
\mathbf{t4} & x \sqsubseteq y \wedge y \sqsubseteq z \rightarrow x \sqsubseteq z
\end{array}$$

We implicitly define *rigidity* of types as rigid (a6), semi-rigid (a8) and anti-rigid (a7), 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)).

$$\begin{array}{l}
\mathbf{a6} \quad \text{Rigid}(t) \leftrightarrow \text{Type}(t) \wedge \forall x(\Diamond(x :: t) \rightarrow \Box(x :: t)) \\
\mathbf{a7} \quad \text{AntiRigid}(t) \leftrightarrow \text{Type}(t) \wedge \forall x(\Diamond(x :: t) \rightarrow \Diamond(\neg x :: t))
\end{array}$$

<sup>2</sup> For the formal specification, see <https://github.com/nemo-ufes/ufo-types>.

- a8**  $\text{SemiRigid}(t) \leftrightarrow \text{Type}(t) \wedge \neg \text{Rigid}(t) \wedge \neg \text{AntiRigid}(t)$
- t5**  $\text{Type}(t) \leftrightarrow \text{Rigid}(t) \vee \text{AntiRigid}(t) \vee \text{SemiRigid}(t)$
- t6**  $\neg \exists x((\text{Rigid}(x) \wedge \text{AntiRigid}(x)) \vee (\text{Rigid}(x) \wedge \text{SemiRigid}(x)) \vee (\text{SemiRigid}(x) \wedge \text{AntiRigid}(x)))$
- t7**  $\neg \exists x, y(\text{Rigid}(x) \wedge \text{AntiRigid}(y) \wedge x \sqsubseteq y)$
- t8**  $\neg \exists x, y(\text{SemiRigid}(x) \wedge \text{AntiRigid}(y) \wedge x \sqsubseteq y)$

On *sortality*, our basic assumption is that every individual necessarily instantiates a kind (a9), and everything necessarily instantiates at most one kind (a10). We implicitly define *sortals* as those types whose instances necessarily instantiate the same 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**  $\text{Individual}(x) \rightarrow \exists k(\text{Kind}(k) \wedge \Box(x :: k))$
- a10**  $\text{Kind}(k) \wedge x :: k \rightarrow \neg \Diamond(\exists z(\text{Kind}(z) \wedge x :: z \wedge z \neq k))$
- a11**  $\text{Sortal}(t) \leftrightarrow \text{Type}(t) \wedge \exists k(\text{Kind}(k) \wedge \Box(\forall x(x :: t \rightarrow x :: k)))$
- a12**  $\text{NonSortal}(t) \leftrightarrow \text{Type}(t) \wedge \neg \text{Sortal}(t)$
- t9**  $\text{Kind}(k) \rightarrow \text{Rigid}(k)$
- t10**  $\text{Kind}(x) \wedge \text{Kind}(y) \wedge x \neq y \rightarrow \Box(\neg \exists z(z :: x \wedge z :: y))$
- t11**  $\text{Kind}(x) \wedge \text{Kind}(y) \wedge x \neq y \rightarrow (\neg(x \sqsubseteq y) \wedge \neg(y \sqsubseteq x))$
- t12**  $\text{Kind}(t) \rightarrow \text{Sortal}(t)$
- t13**  $\text{Sortal}(x) \rightarrow \exists k(\text{Kind}(k) \wedge x \sqsubseteq k)$
- t14**  $\neg \exists x, y, z(\text{Kind}(y) \wedge \text{Kind}(z) \wedge y \neq z \wedge x \sqsubseteq y \wedge x \sqsubseteq z)$
- t15**  $\neg \exists x, y(\text{NonSortal}(x) \wedge \text{Sortal}(y) \wedge x \sqsubseteq y)$
- t16**  $(\text{NonSortal}(t) \wedge x :: t) \rightarrow (\exists s(\text{Sortal}(s) \wedge s \sqsubseteq t \wedge x :: s) \vee \exists n, s(\text{NonSortal}(n) \wedge \text{Sortal}(s) \wedge s \sqsubseteq n \wedge t \sqsubseteq n \wedge x :: s))$

Regarding the leaves of the taxonomy of 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 antirigid sortals (a16). Semi rigid sortals are those that are semirigid and sortal (a17). Categories are those types that are rigid and non-sortals (a18). Mixins are those types that are semirigid and non-sortals (a19). Phase-mixins and role-mixins are disjoint (a20), and together encompass all antirigid non-sortals (a21). Let  $\mathcal{L}_T$  be the set of the leaf categories of the UFO taxonomy of types  $\{\text{Kind}, \text{SubKind}, \text{Role}, \text{Phase}, \text{SemiRigidSortal}, \text{RoleMixin}, \text{PhaseMixin}, \text{Category}, \text{Mixin}\}$ , it follows that these leaf categories are pairwise disjoint (t17) and complete (t18).

- a13**  $\neg \exists t(\text{Kind}(t) \wedge \text{SubKind}(t))$

- a14**  $\text{Kind}(t) \vee \text{SubKind}(t) \leftrightarrow \text{Rigid}(t) \wedge \text{Sortal}(t)$   
**a15**  $\neg \exists t (\text{Phase}(t) \wedge \text{Role}(t))$   
**a16**  $\text{Phase}(t) \vee \text{Role}(t) \leftrightarrow \text{AntiRigid}(t) \wedge \text{Sortal}(t)$   
**a17**  $\text{SemiRigidSortal}(t) \leftrightarrow \text{SemiRigid}(t) \wedge \text{Sortal}(t)$   
**a18**  $\text{Category}(t) \leftrightarrow \text{Rigid}(t) \wedge \text{NonSortal}(t)$   
**a19**  $\text{Mixin}(t) \leftrightarrow \text{SemiRigid}(t) \wedge \text{NonSortal}(t)$   
**a20**  $\neg \exists t (\text{PhaseMixin}(t) \wedge \text{RoleMixin}(t))$   
**a21**  $\text{PhaseMixin}(t) \vee \text{RoleMixin}(t) \leftrightarrow \text{AntiRigid}(t) \wedge \text{NonSortal}(t)$   
**t17**  $\bigwedge_{i,j \in \mathcal{L}_T, i \neq j} i(t) \rightarrow \neg j(t)$       **t18**  $\text{Type}(x) \leftrightarrow \bigvee_{i \in \mathcal{L}_T} i(x)$

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  $\{\text{Substantial}, \text{Relator}, \text{Mode}, \text{Quality}\}$ , it is a theorem that these leaf categories partition Endurant (t19), (t20).

- a22**  $\text{Endurant}(x) \rightarrow \text{Individual}(x)$   
**a23**  $\text{Substantial}(x) \vee \text{Moment}(x) \leftrightarrow \text{Endurant}(x)$   
**a24**  $\neg \exists x (\text{Substantial}(x) \wedge \text{Moment}(x))$   
**a25**  $\text{Relator}(x) \vee \text{IntrinsicMoment}(x) \leftrightarrow \text{Moment}(x)$   
**a26**  $\neg \exists x (\text{Relator}(x) \wedge \text{IntrinsicMoment}(x))$   
**a27**  $\text{Mode}(x) \vee \text{Quality}(x) \leftrightarrow \text{IntrinsicMoment}(x)$   
**a28**  $\neg \exists x (\text{Mode}(x) \wedge \text{Quality}(x))$   
**t19**  $\bigwedge_{i,j \in \mathcal{L}_e, i \neq j} i(t) \rightarrow \neg j(t)$   
**t20**  $\text{Endurant}(x) \leftrightarrow \text{Substantial}(x) \vee \text{Relator}(x) \vee \text{Mode}(x) \vee \text{Quality}(x)$

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

- a29**  $\bigwedge_{(i,j) \in \mathcal{P}_E} i(t) \leftrightarrow \text{Type}(t) \wedge \Box(\forall x(x :: t \rightarrow j(x)))$   
**t21**  $\bigwedge_{i,j \in \{\text{SubstantialType}, \text{RelatorType}, \text{ModeType}, \text{QualityType}\}, i \neq j} i(x) \rightarrow \neg j(x)$

Kinds are also specialized according to the ontological nature of their instances. Let  $\mathcal{P}_K$  be the set of pairs  $\{(\text{SubstantialKind}, \text{SubstantialType}); (\text{RelatorKind}, \text{RelatorType}); (\text{ModeKind}, \text{ModeType}); (\text{QualityKind}, \text{QualityType})\}$ . 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 enduring sortal is a type in  $\mathcal{L}_{ES} = \{\text{SubstantialKind}, \text{RelatorKind}, \text{ModeKind}, \text{QualityKind}, \text{SubKind}, \text{Phase}, \text{Role}, \text{SemiRigidSortal}\}$  (t23); and that some sortals specialize specific kinds (t24).

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

$$\mathbf{t22} \quad \diamond(\exists k((\text{SubstantialKind}(k) \vee \text{RelatorKind}(k) \vee \text{ModeKind}(k) \vee \text{QualityKind}(k)) \wedge x :: k)) \rightarrow \text{Endurant}(x)$$

$$\mathbf{a31} \quad \text{Endurant}(x) \rightarrow \diamond(\exists k((\text{SubstantialKind}(k) \vee \text{RelatorKind}(k) \vee \text{ModeKind}(k) \vee \text{QualityKind}(k)) \wedge x :: k))$$

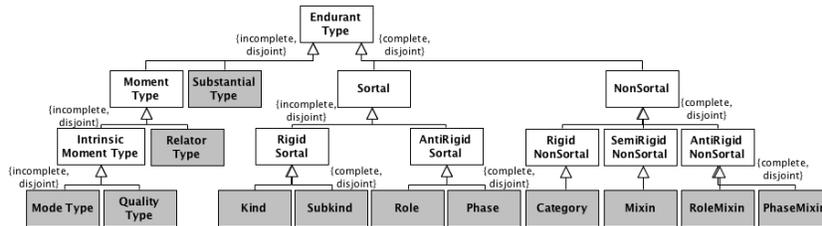
$$\mathbf{t23} \quad \text{EndurantType}(x) \wedge \text{Sortal}(x) \rightarrow \bigvee_{i \in \mathcal{L}_{ES}} i(x)$$

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

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

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

The results of this section are summarized in Figure 3. When contrasting this figure with Figure 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 [6]. 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 [8], as also confirmed by the empirical analysis of OntoUML models.



**Fig. 3.** Proposed taxonomy of enduring types in UFO.

## 4 Towards a new UML Profile for Modeling Endurant Taxonomic Structures

OntoUML is a 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 ontology. Additionally, a number of *semantically motivated syntactic constraints* [3] 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 this section we provide a new UML profile (*lightweight extension*) for OntoUML that reflects the taxonomy of enduring types previously discussed and presented in Fig. 3. Only the elements in gray of the profile presented in Fig. 4 represent concrete stereotypes.

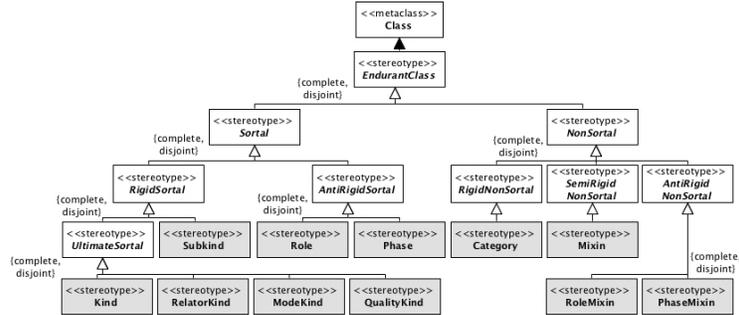


Fig. 4. OntoUML profile.

In the current version of OntoUML, the only stereotype for ultimate sortals (i.e. kinds) is «kind», which decorates substantial types that are ultimate sortals. Since the language was silent about modal properties of other types of endurants the stereotypes «relator», «mode» and «quality» would only carry information about the nature of the instances of that given moment type. In order to account for modal properties of both substantial and moment types, we explore the new taxonomy of enduring types proposed in Section 3 and define four stereotypes for identifying ultimate sortals according to the specific nature of its instances: «kind» represents a ultimate sortal whose instances are substantials (e.g., *Person, Organization, House*); «relatorKind» represents a ultimate sortal whose instances are relators (e.g., *Marriage, Employment, Enrollment*); «modeKind» represents an ultimate sortal whose instances are modes (e.g., *Headache, Intention, Goal*); and «qualityKind» represents a ultimate sortal whose instances are qualities (e.g., *Color, Weight, Height*). Classes decorated with a stereotype from this set not only provide the principle of identity to their instances, but also identify their nature. Notice that «kind» keeps the same semantics from the current profile, but «relator», «mode» and «quality» are discontinued in the new

one. Furthermore, the additional stereotypes of this profile {«subkind», «role», «phase», «category», «mixin», «roleMixin», «phaseMixin»} have direct counterparts in UFO (see Fig. 3) and each of these stereotypes reflect the properties of its counterpart into the decorated class. In Fig. 5 and Fig. 6 we present examples of their application. These examples are not intended to defend the particular modeling choices therein, but rather to elucidate language application.

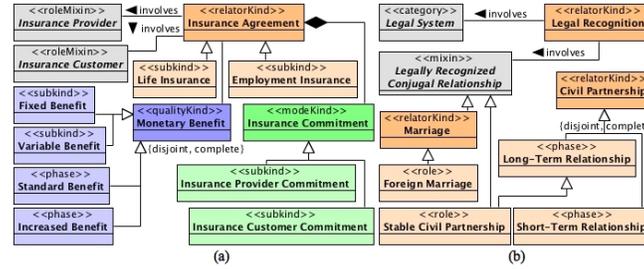
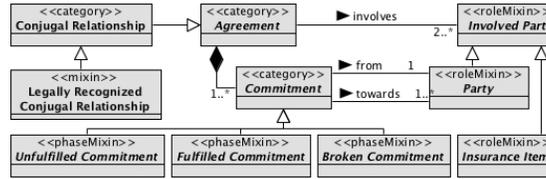


Fig. 5. Examples of sortal moment types.

From (a13) and (a14) we have that every class representing a sortal specializes a unique kind. Therefore, other classes decorated with sortal stereotypes may rely on their kinds to define the nature of their instances. For instance, the stereotype «subkind», which represents a rigid sortal that inherits its principle of identity from a kind, may decorate classes that apply to any nature (e.g., subkinds of relator types). Fig. 5 contains examples on the domain of agreements (e.g., insurance agreements, civil partnerships, marriages) and related sortals represented in the proposed profile. In this example, *Life Insurance* and *Employment Insurance* are rigid types that inherit their principle of identity and their nature from *Insurance Agreement*, and thus, are decorated as «subkind» and represent relator types. Likewise, *Insurance Provider Commitment* and *Insurance Customer Commitment*, which represent commitments of the provider and customer involved in an insurance agreement, respectively, are rigid sortal mode types whose instances constitute *Insurance Agreements*. Our conceptualization may also include *Monetary Benefit* as a «qualityKind» that represents the amount of money the customer may receive for her insurance, amount that may be fixed *a priori* or variable according to the conditions of the agreement.

Moreover, «role» and «phase» represent anti-rigid sortals that are externally dependent, in the case of roles, or independent, in the case of phases. Fig. 5.(b) includes examples of phases and roles of moment types that are covered by the new profile. *Foreign Marriage*, for instance, can be defined as a «role» played by a marriage involved in a legal recognition relationship with a foreign legal system. *Civil Partnership*, another sort of relationship, may be classified into phases of short-term and long-term relationship. In legal systems where long-term relationships are legally recognized, acquiring marriage-like status, we also have the role *Stable Civil Partnership*. As an example of «phase» of quality types,

*Standard Benefit* and *Increased Benefit* can represent benefits of an insurance which may change its prize value according to the clauses of the related contract.



**Fig. 6.** Examples of non-sortal universals.

Fig. 6 presents examples of non-sortal moment types, i.e., types of moments whose instances follow different principles of identity. Here, *Agreement* represents a general category of agreements (e.g., conjugal relationships, insurance agreements). *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 an 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 the sortals, this profile does not explicitly capture the nature of instances of non-sortal classes. Accordingly, non-sortal classes may classify not only endurants of distinct principles of identity, but also 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 or a contract), mixing instances of different basic ontological categories.

Table 1 summarizes the *semantically motivated syntactic constraints* regarding this new profile. The constraints discussed here are limited to those related to taxonomies of endurant types. Along with this new OntoUML profile, the revision of the language includes the development of computational support for ODCM. In that spirit, we have developed a plug-in for the Visual Paradigm UML CASE tool<sup>3</sup>, which contemplates the aforementioned stereotypes and allows the verifications of the related syntactic constraints over models, thus, supporting the user for consistent representation of the intended conceptualizations. In case of constraint violation, the plug-in presents a log of the inconsistencies identified and the involved entities.

<sup>3</sup> The OntoUML plug-in for Visual Paradigm is available at <https://github.com/nemo-ufes/OntoUML-2.0-for-Visual-Paradigm>.

**Table 1.** OntoUML constraints for taxonomies of enduring universals.

Constraints
From (t25) and (t26), every class representing an enduring type must be decorated with exactly one stereotype from the set {«kind», «relatorKind», «modeKind», «qualityKind», «subkind», «role», «phase», «category», «mixin», «roleMixin», «phaseMixin»}. Semi-rigid sortals are excluded from the profile (see sec.3).
From (t24), every class representing an enduring sortal that is not a kind (including «subkind», «role», «phase») specializes a class decorated with a stereotype from the set {«kind», «relatorKind», «modeKind», «qualityKind»}.
From (t11), a class representing a kind cannot specialize another kind.
From (t14), a class cannot specialize more than one kind.
From (t7), a class representing a rigid type {«kind», «relatorKind», «modeKind», «qualityKind», «subkind», «category»} cannot specialize a class representing an anti-rigid type {«role», «phase», «roleMixin», «phaseMixin»}.
From (t8), a class representing a semi-rigid type {«mixin»} cannot specialize a class representing an anti-rigid type {«role», «phase», «roleMixin», «phaseMixin»}.
From (t15), a class representing a non-sortal {«category», «mixin», «roleMixin», «phaseMixin»} cannot specialize a class representing a sortal one {«kind», «relatorKind», «modeKind», «qualityKind», «subkind», «role», «phase»}.
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 $S$ .

## 5 Final Considerations

In this paper, we make a contribution 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 a number of 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 philosophical literature (*claim to ontological adequacy*). For example, Moltmann [10] uses the notion of *variable tropes* (moments that can change while maintaining identity) to address a number of fundamental phenomena in cognition and language. Additionally, this formal theory has been checked for its consistency using theorem provers (*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 foundation 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 [8]. 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 constraint proposed by this theory; a computational tool implementing this profile.

The work developed here focuses exclusively on enduring types and taxonomic relations. In an extension of this work, we intend to address other relations involving durants and enduring types (e.g., relational dependence for roles and role mixins, existential dependence for moment types, the foundation between relators and events [8]). More broadly, the work presented is part of a research program aimed at addressing a fuller evolution of UFO and OntoUML.

## Acknowledgments

We thank Nicola Guarino for the fruitful discussions; and CNPq, CAPES, and FAPES for funding. The third author was supported by the grant N° 71024352.

## References

1. Blums, I., Weigand, H.: Financial reporting by a shared ledger. In: 8th International Workshop on Formal Ontologies Meet Industry (FOMI) (2017)
2. do Carmo, A.P., Zamperini, T., de Mello, M.R., de Castro Leal, A.L., Garcia, A.S.: Ontologia das coisas para espaços inteligentes baseados em visão computacional. In: 9th Brazilian Ontology Research Seminar (2017)
3. Carvalho, V.A., Almeida, J.P.A., Guizzardi, G.: Using reference domain ontologies to define the real-world semantics of domain-specific languages. In: Proc. of the 26th CAISE. pp. 488–502. Springer (2014)
4. de Carvalho, V.A., Almeida, J.P.A., Fonseca, C.M., Guizzardi, G.: Multi-level ontology-based conceptual modeling. *Data Knowl. Eng.* **109**, 3–24 (2017)
5. Fitting, M., Mendelsohn, R.L.: First-order modal logic, vol. 277. Springer Science & Business Media (2012)
6. Guarino, N., Guizzardi, G.: “We need to discuss the relationship”: revisiting relationships as modeling constructs. In: Proc. of the 27th CAISE (2015)
7. Guarino, N., Welty, C.A.: An overview of ontoclean. In: Staab, S., Studer, R. (eds.) *Handbook on Ontologies*, pp. 201–220. Springer (2009)
8. Guizzardi, G.: Ontological foundations for structural conceptual models. *Telematica Instituut / CTIT* (2005)
9. Guizzardi, G., Wagner, G., Almeida, J.P.A., Guizzardi, R.S.: Towards ontological foundations for conceptual modeling: the unified foundational ontology (ufo) story. *Applied ontology* **10**(3-4), 259–271 (2015)
10. Moltmann, F.: Events, tropes and truthmaking. *Philosophical Studies* (2007)
11. Rybala, Z.: Towards OntoUML for Software Engineering: Transformation of OntoUML into Relational Databases. Czech Technical University in Prague (2017)
12. Verdonck, M.: Ontology-Driven Conceptual Modeling: Model Comprehension, Ontology Selection, and Method Complexity. PhD thesis submitted to the Applied Economics Program of Ghent University, Belgium (2018)
13. Verdonck, M., Gailly, F.: Insights on the use and application of ontology and conceptual modeling languages in ontology-driven conceptual modeling. In: *Conceptual Modeling - 35th International Conference, ER 2016, Gifu*. pp. 83–97 (2016)
14. Zamborlini, V., Betti, A., van den Heuvel, C.: Toward a core conceptual model for (im)material cultural heritage in the golden agents project. In: 1st International Workshop on Understanding Events Semantics in Cultural Heritage (2017)