

# Using the Unified Foundational Ontology (UFO) as a Foundation for General Conceptual Modeling Languages

Giancarlo Guizzardi<sup>1,2</sup>, Gerd Wagner<sup>3</sup>

<sup>1</sup>Federal University of Espirito Santo (UFES), Vitoria-ES, Brazil

<sup>2</sup>Laboratory for Applied Ontology (ISTC-CNR), Trento, Italy

<sup>3</sup>Brandenburg University of Technology at Cottbus, Cottbus, Germany

Emails: guizzardi@loa-cnr.it; G.Wagner@tu-cottbus.de

**Abstract.** In recent years, there has been a growing interest in approaches that employ ontological models as theoretical tools for analyzing and improving conceptual modeling languages. In this paper we present a philosophically and cognitively well-founded formal ontology which has been developed with the special purpose of serving as a foundation for general conceptual modeling languages. Furthermore, we demonstrate how this foundational ontology named the Unified Foundational Ontology (UFO) has been used to evaluate and redesign the metamodel of the Unified Modeling Language (UML) for the purpose of conceptual modeling.

## 1 Introduction

In recent years, there has been a growing interest in the use of foundational ontologies (also known as upper level, or top-level ontologies) for: (i) evaluating conceptual modeling languages; (ii) developing guidelines for their use; (iii) providing real-world semantics for their modeling constructs. In this paper, we present a fragment of a philosophically and cognitively well-founded reference ontology named *UFO (Unified Foundational Ontology)*. UFO started as a unification of the GFO (Generalized Formalized Ontology) [5] and the Top-Level ontology of universals underlying OntoClean (<http://www.ontoclean.org>). However, as shown in [1], there are a number of problematic issues related the specific objective of developing ontological foundations for general conceptual modeling languages (e.g., EER, UML, ORM) which were not covered in a satisfactory manner by existing foundational ontologies such as GFO, DOLCE or OntoClean. For this reason, UFO has been developed into a full-blown reference ontology of endurants based on a number of theories from Formal Ontology, Philosophical Logics, Philosophy of Language, Linguistics and Cognitive Psychology. This ontology is presented in depth and formally characterized in [1]. In section 2, we discuss the main categories comprising UFO.

Furthermore, we demonstrate in this paper how this ontology can be used in the design and evaluation of conceptual modeling languages. In section 3, we present a general ontology-based framework that can be used to systematically assess the suitability of an artificial modeling language to model phenomena in a given domain. In

particular, this framework focuses on two properties of modeling languages [1]: (i) *domain appropriateness*, which refers to truthfulness of a language to a given domain in reality; (ii) *comprehensibility appropriateness*, which refers to the pragmatic efficiency of a language to support communication, domain understanding and reasoning in that domain.

In section 4 and 5, we employ UFO and the framework of section 3 to analyze and redesign the 2.0 version of the metamodel of the Unified Modeling Language (UML) [12]. The fact that UML is a *de facto* standard considered in several sub-fields of computer science (e.g., software and domain engineering, database and information systems design) counts in favor of the practicality and relevance of this approach. Section 6 presents some final considerations of the article.

## 2 The Unified Foundational Ontology (UFO)

In the sequel, we restrict ourselves to a fragment of UFO, depicted in the figure 2. Moreover, due to space limitations and the focus of the paper we present the ontological categories comprising UFO superficially. For an in depth presentation and corresponding formalization, one should refer to [1].

### 2.1 The Core Categories: Object-Object Universal, Moment-Moment Universal

A fundamental distinction in this ontology is between the categories of *Particular* and *Universal*. Particulars are entities that exist in reality possessing a unique identity. Universals, conversely, are pattern of features, which can be realized in a number of different particulars. The core of this ontology exemplifies the so-called *Aristotelian ontological square* or what is termed a “*Four-Category Ontology*” [8] comprising the category pairs *Object-Object Universal*, *Moment-Moment Universal*. From a metaphysical point of view, this choice allows for the construction of a parsimonious ontology, based on the primitive and formally defined notion of *existential dependence*: We have that a particular  $x$  is *existentially dependent* (**ed**) on another particular  $y$  iff, as a matter of necessity,  $y$  must exist whenever  $x$  exists. Existential dependence is a modally constant relation, i.e., if  $x$  is dependent on  $y$ , this relation holds between these two specific particulars in all possible worlds in which  $x$  exists.

The word *Moment* is derived from the german *Momente* in the writings of E. Husserl and it denotes, in general terms, what is sometimes named *trope*, *abstract particular*, *individual accident*, or *property instance*. Thus, in the scope of this work, the term bears no relation to the notion of time instant in colloquial language. Typical examples of moments are: a color, a connection, an electric charge, a social commitment. An important feature that characterizes all *moments* is that they can only exist in other particulars (in the way in which, for example, electrical charge can exist only in some conductor). To put it more technically, we say that moments are *existentially dependent* on other particulars. Existential dependence can also be used to differentiate intrinsic and relational moments: *intrinsic moments* are dependent of one single particular (e.g., color, a headache, a temperature); *relators* depend on a plurality of particulars (e.g., an employment, a medical treatment, a marriage). A special type of

existential dependence relation that holds between a moment  $x$  and the particular  $y$  of which  $x$  depends is the relation of *inherence* ( $i$ ). Thus, for a particular  $x$  to be a moment of another particular  $y$ , the relation  $i(x,y)$  must hold between the two. For example, inherence glues your smile to your face, or the charge in a specific conductor to the conductor itself. Here, we admit that moments can inhere in other moments. Examples include the individualized time extension, or the graveness of a particular symptom. The infinite regress in the inherence chain is prevented by the fact that there are individuals that cannot inhere in other individuals, namely, *objects*.

*Objects* are particulars that possess (direct) spatial-temporal qualities and that are founded on matter. Examples of objects include ordinary entities of everyday experience such as an individual person, a dog, a house, a hammer, a car, Alan Turing and The Rolling Stones but also the so-called *Fiat Objects* such as the North-Sea and its proper-parts, postal districts and a non-smoking area of a restaurant. In contrast with moments, objects do not inhere in anything and, as a consequence, they enjoy a higher degree of independence. To state this precisely we say that: an object  $x$  is *independent* of all other objects which are disjoint from  $x$ , i.e., that do not share a common part with  $x$ , where  $\text{independent}(x,y) =_{\text{def}} \neg\text{ed}(x,y) \wedge \neg\text{ed}(y,x)$ . This definition excludes the dependence between an object and its *essential* and *inseparable parts* [1], and the obvious dependence between an object and its essential moments.

To complete the Aristotelian Square, we consider here the categories of *object universal* and *moment universal*. We use the term universal here in a broader sense without making any *a priori* commitment to a specific theory of universals. A universal thus can be considered here simply as something (i) which can be predicated of other entities and (ii) that can potentially be represented in language by *predicative terms*. We also use the relation  $::$  of classification between particulars and universals. Object universals classify objects and moment universals classify moments. Examples of the former include Apple, Planet and Person. Examples of the latter include Color, Electric Charge and Headache. This distinction is also present in Aristotle's original differentiation between what is *said of a subject* (*de subjecto dici*), denoting classification and what is *exemplified in a subject* (*in subjecto est*), denoting inherence. Thus, the linguistic difference between the two meanings of the copula "is" reflects an ontological one. For example, the ontological interpretation of the sentence "Jane is a Woman" is that the Object Jane is classified by the Object kind Woman. However, when saying that "Jane is tall" or "Jane is laughing" we mean that Jane *exemplifies* the moment universal Tall or Laugh, by virtue of her specific height or laugh. Finally, we define the relation of *characterization* between moment universals and the particulars that exemplify them: a moment universal  $M$  characterizes a universal  $U$  iff every instance of  $U$  exemplifies  $M$ . The categories of object, moment, object universal and moment universal as well as the relations of classification, inherence, exemplification and characterization are organized in terms of the so-called Aristotelian Square in figure 1.

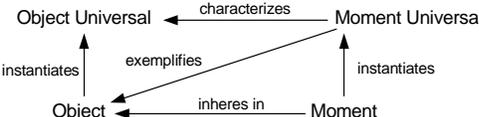


Fig 1. The Aristotelian Square

## 2.2 Qualities, Qualia and Modes

An attempt to model the relation between intrinsic moments and their representation in human cognitive structures is presented in the theory of *conceptual spaces* introduced in [3]. The theory is based on the notion of *quality dimension*. The idea is that for several perceivable or conceivable moment universals there is an associated quality dimension in human cognition. For example, height and mass are associated with one-dimensional structures with a zero point isomorphic to the half-line of nonnegative numbers. Other properties such as color and taste are represented by multi-dimensional structures.

In [3], the author distinguishes between *integral* and *separable* quality dimensions: “certain quality dimensions are integral in the sense that one cannot assign an object a value on one dimension without giving it a value on the other. For example, an object cannot be given a hue without giving it a brightness value (...) Dimensions that are not integral are said to be separable, as for example the size and hue dimensions.” He then defines a *quality domain* as “a set of integral dimensions that are separable from all other dimensions” and a *conceptual space* as a “collection of one or more domains” (ibid.). Finally, he defends that the notion of conceptual space should be understood literally, i.e., quality domains are endowed with certain geometrical structures (topological or ordering structures) that constrain the relations between its constituting dimensions. In [3], the perception or conception of an intrinsic moment can be represented as a point in a quality domain. Following [4], this point is named here a *quale*.

An example of a quality domain is the set of integral dimensions related to color perception. A color quality  $c$  of an apple  $a$  takes its value in a three-dimensional color domain constituted of the dimensions hue, saturation and brightness. The geometric structure of this space (the *color splinter* [3]) constrains the relation between some of these dimensions. In particular, saturation and brightness are not totally independent, since the possible variation of saturation decreases as brightness approaches the extreme points of black and white, i.e., for almost black or almost white, there can be very little variation in saturation. A similar constraint could be postulated for the relation between saturation and hue. When saturation is very low, all hues become similarly approximate to grey.

We adopt in this work the term *quality structures* to refer to quality dimensions and quality domains, and we define the formal relation of *association* between a quality structure and a moment universal. Additionally, we use the terms *quality universals* for those intrinsic moment universals that are *associated with* a quality structure, and the term *quality* for a moment classified under a quality universal. We also assume that quality structures are always associated with a unique quality universal, i.e., a quality structure associated with the universal Weight cannot be associated with the universal Color.

Following [4], we take that whenever a quality universal  $Q$  is related to a quality domain  $D$ , then for every particular quality  $x::Q$  there are *indirect qualities* inhering in  $x$  for every quality dimension associated with  $D$ . For instance, for every particular quality  $c$  instance of Color there are quality particulars  $h, s, b$  which are instances of quality kinds Hue, Saturation and Brightness, respectively, and that inhere in  $c$ . The

qualities *h*, *s*, *b* are named *indirect qualities* of *c*'s bearer. Qualities such as *h*, *s*, *b* are named *simple qualities*, i.e., qualities which do not bear other qualities. In contrast, a quality such as *c*, is named a *complex quality*. Since the qualities of a complex quality  $x::Q$  correspond to the quality dimensions of the quality domain associated with *Q*, then we have that no two distinct qualities inhering a complex quality can be of the same type. For the same reason, since there are not multidimensional quality dimensions, we have that complex qualities can only bear simple qualities. Moreover, we use the predicate *qualeOf(x,y)* to represent the formal relation between a quality particular *y* and its quale *x*.

Finally, we make a distinction between qualities and another sort of intrinsic moment named here *modes*. Modes are moments whose universals are not directly related to quality structures. In [3], the author makes the following distinction between what he calls *concepts* and *properties* (which at first could be thought to correspond to the distinction between Object and Moment universals, respectively): “Properties...form as special case of concepts. I define this distinction by saying that a *property* is based on *single domain*, while a *concept* may be based on *several domains*”. We claim, however, that only moment universals that are conceptualized w.r.t. a single domain, i.e., quality universals, correspond to properties in the sense of [3]. There are, nonetheless, moment universals that as much as object universals can be conceptualized in terms of multiple separable quality dimensions. Examples include beliefs, desires, intentions, perceptions, symptoms, skills, among many others. Like objects, modes can bear other moments, and each of these moments can refer to separable quality dimensions. However, since they are moments, differently from objects, modes are necessarily existentially dependent of some particular.

### 2.3. Relations, Relators and Qua Individuals

*Relations* are entities that glue together other entities. In the philosophical literature, two broad categories of relations are typically considered, namely, *material* and *formal* relations [5,6]. Formal relations hold between two or more entities directly, without any further intervening particular. In principle, the category of formal relations includes those relations that form the mathematical superstructure of our framework including existential dependence (*ed*), inherence (*i*), *part-of* (<), *subset-of*, *instantiation*, *characterization*, *exemplification*, among many others not discussed here [1]. We name these relations here *basic formal relations* [5] or *internal relations* [7]. In this case, in conformance with [7] we deem the tie (or nexus) between the relata as non-analyzable.

However, we also classify as formal those domain relations that exhibit similar characteristics, i.e., those relations of *comparison* such as *is taller than*, *is older than*, *knows more greek than*. We name these relations *comparative formal relations*. As pointed out in [6], the entities that are immediate relata of such relations are not objects but intrinsic moments. For instance, the relation *heavier-than* between two atoms is a formal relation that holds directly as soon as the relata (atoms) are given. The truth-value of a predicate representing this relation depends solely on the atomic number (a quality) of each atom and the material content of *heavier-than* is as it were distributed between the two relata. Moreover, to quote Mulligan and Smith, “once the

distribution has been effected, the two relata are seen to fall apart, in such a way that they no longer have anything specifically to do with each other but can serve equally as terms in a potentially infinite number of comparisons”.

*Material relations*, conversely, have material structure of their own and include examples such as *working at*, *being enrolled at*, and *being connected to*. Whilst a formal relation such as the one between Paul and his knowledge  $x$  of Greek holds directly and as soon as Paul and  $x$  exist, for a material relation of *being treated in* between Paul and the medical unit  $MU_1$  to exist, another entity must exist which *mediates* Paul and  $MU_1$ . We name these entities *relators*. Relators are particulars with the power of connecting entities. For example, a medical treatment connects a patient with a medical unit; an enrollment connects a student with an educational institution; a covalent bond connects two atoms. The notion of relator (relational moment) is supported by several works in the philosophical literature [5,6,8] and, the position advocated here is that they play an important role in answering questions of the sort: what does it mean to say that John is married to Mary? Why is it true to say that Bill works for Company X but not for Company Y?

An important notion for the characterization of relators (and, hence, for the characterization of material relations) is the notion of *foundation*. Foundation can be seen as a type of *historical dependence* [9], in the way that, for example, an instance of *being kissed* is founded on an individual *kiss*, or an instance of *being punched by* is founded on an individual *punch*, an instance of *being connected to* between airports is founded on a particular flight connection. Suppose that John *is married to* Mary. In this case, we can assume that there is a particular relator (relational moment)  $m_1$  of type *marriage* that mediates John and Mary. The foundation of this relator can be, for instance, a wedding event or the signing of a social contract between the involved parties. In other words, for instance, a certain event  $e_1$  in which John and Mary participate can create a particular marriage  $m_1$  which existentially depends on John and Mary and which mediates them. The event  $e_1$  in this case is the foundation of relator  $m_1$  and,  $m_1$  is the so-called truthmaker of the propositions “John is married to Mary”.

Using this example, we can further elaborate on the nature of the relator  $m_1$ . There are many moments that John acquires by virtue of being married to Mary. For example, imagine all the legal responsibilities that John has in the context of this relation. These newly acquired properties are intrinsic moments of John which, therefore, inhere and are existentially dependent on him. However, these moments also depend on the existence of Mary. We name this type of moment *externally dependent moment*, i.e., externally dependent moments are intrinsic moments that inhere in a single particular but that are existentially dependent on (possibly a multitude of) other particulars: a moment  $x$  is externally dependent iff it is existentially dependent of a particular which is *independent* (in the technical sense of 2.1) of its bearer.

In the case of a material externally dependent moment  $x$  there is always a particular *external* to its bearer (i.e., which is not one of its parts or intrinsic moments), which is the foundation of  $x$ . Again, in the given example, we can think of a certain event  $e_1$  (wedding event or signing of social contract) in which both John and Mary participate and which founds the existence of these externally dependent moments inhering in John. Now, we can define a particular that bears all externally dependent moments of John that share the same external dependencies and the same foundation. We term this

particular a *qua individual* [10]. Qua individuals are, thus, treated here as a special type of *complex externally dependent modes*. In this case, the complex mode inhering in John that bears all responsibilities that John acquires by virtue of a given wedding event can be named *John-qua-husband*.

To continue with the same example, we can think about another qua individual *Mary-qua-wife* which is a complex mode bearing all responsibilities that Mary acquires by virtue of the same foundation and that albeit inhering in Mary are also existentially dependent on John. The qua individuals *John-qua-husband* and *Mary-qua-wife* are existentially dependent on each other. Now, we can define an aggregate  $m_1$  composed of these two qua individuals that share the same foundation, i.e., (*John-qua-husband* <  $m_1$ ) and (*Mary-qua-wife* <  $m_1$ ). In this example,  $m_1$  is exactly the instance of the relational property *marriage* that mediates John and Mary and that makes true propositions such as “John is married to Mary”, “Mary is married to John”, “John is the husband of Mary”, and “Mary is the wife of John”.

In this example, a particular instance of the relational property marriage (i.e., a particular marriage relator) is the sum of all instantiated responsibilities that the involved parties acquire by virtue of a common foundation. In general, a relator can be defined as the aggregation of a number of qua individuals that share the same foundation. A relator is said to mediate (or connect) the relata of a material relation. Formally we have that: let  $x$ ,  $y$  and  $z$  be three distinct individuals such that (a)  $x$  is a relator; (b)  $z$  is a qua individual and  $z$  is part of  $x$ ; (c)  $z$  inheres in  $y$ . In this case, we say that  $x$  mediates  $y$ , symbolized by  $m(x,y)$ . Additionally, we require that a relator mediates at least two distinct particulars. Again, using the example above, we say that the particular relator marriage  $m_1$  mediates the objects John and Mary and, for this reason, we can say that John and Mary are married to each other.

Analogous to the relation of characterization, we define a relation of *mediation* that can obtain between a set of object universal and a relator universal in the following way: If a relator universal  $U_R$  mediates the object universals  $S_1...S_n$ , then every instance of  $U_R$  is existentially dependent on a plurality of entities, namely, particular instances of  $S_1...S_n$ . Relator universals constitute the basis for defining material relations  $R$ . Material relations are themselves universals whose instances are n-tuples of particulars. We define the formal relation of derivation *derivation*( $R, U_R$ ) holding between a relator universal  $U_R$  and a material relation  $R$  such that a n-tuple  $\langle x_1...x_n \rangle$  instantiates  $R$  iff there is a relator  $r:: U_R$  such that  $r$  mediates every  $x_i$ . To employ once more the example above, we have that as  $\langle \text{John, Mary} \rangle$  is an instance of both *married to* and *is the husband of*, and  $\langle \text{Mary, John} \rangle$  is an instance of both *married to* and *is the wife of* because there is an individual marriage relator  $m_1$  that mediates John and Mary.

## 2.4 Object Universals

Here we considered a fundamental distinction in the category of object Universals, namely, the one between *Sortal* and *Mixin Universals*. Whilst all universals carry a *principle of application*, only sortals carry a *principle of identity* for their instances. A principle of application is a principle for which we can judge whether a particular is

an instance of that universal. In contrast, a principle of identity is a principle for which we can judge whether two particulars are the same. As an illustration of this point, contrast the two universals Apple and Red<sup>1</sup> instantiated by two particulars  $x$  and  $y$ : both universals supply a principle for which we can judge whether  $x$  and  $y$  are classified under those types (i.e., whether they are Apples, or Reds). However, only Apple supplies a principle for which we decide whether  $x$  and  $y$  are the same (i.e., merely knowing that  $x$  and  $y$  are both red gives no clue to decide whether or not  $x=y$ ).

Within the category of sortal universals, we make a further distinction based on the formal notions of *rigidity* and *anti-rigidity*: A universal  $U$  is rigid if for every instance  $x$  of  $U$ ,  $x$  is necessarily (in the modal sense) an instance of  $U$ . In other words, if  $x$  instantiates  $U$  in a given world  $w$ , then  $x$  must instantiate  $U$  in every possible world  $w'$ . In contrast, a universal  $U$  is anti-rigid if for every instance  $x$  of  $U$ ,  $x$  is *possibly* (in the modal sense) not an instance of  $U$ . In other words, if  $x$  instantiates  $U$  in a given world  $w$ , then there must be a possible world  $w'$  in which  $x$  does not instantiate  $U$ . A sortal universal which is rigid is named here a **Kind**. In contrast, an anti-rigid sortal universal is termed here a **Phased-Sortal**. The prototypical example highlighting the modal distinction between these two categories is the difference between the Kind Person and the Phase-Sortals Student and Adolescent instantiated by the particular John in a given circumstance. Whilst John can cease to be a Student and Adolescent (and there were circumstances in which John was not one), he cannot cease to be a Person. In other words, while the instantiation of the phase-sortals Student and Adolescent has no impact on the identity of a particular, if a particular ceases to instantiate the universal Person, then she ceases to exist as the same particular.

John can move in and out of the Student universal, while being the same particular, i.e. without losing his identity. This is because the principle of identity that applies to instances of Student and, in particular, that can be applied to John, is the one which is supplied by the kind Person of which the phase-sortal Student is a subtype. This is always the case with Phased-Sortals, i.e., for every phased-sortal PS, there is a unique ultimate kind  $K$ , such that: (i) PS is a specialization of  $K$ ; (ii)  $K$  supplies the unique principle of identity obeyed by the instances of PS. If PS is a phased-sortal and  $K$  is the kind specialized by PS, there is a *specialization condition*  $\phi$  such that  $x$  is an instance of PS iff  $x$  is an instance of  $K$  that satisfies  $\phi$ . A further clarification on the different types of specialization conditions allows us to distinguish between two different types of phased-sortals: **Phases** and **Roles**. Phases constitute possible stages in the history of a particular. Examples include: (a) Alive and Deceased: as possible stages of a Person; (b) Catterpillar and Butterfly of a Lepidopteran; (c) Town and Metropolis of a City; (d) Boy, Male Teenager and Adult Male of a Male Person. Roles differ from phases with respect to the specialization condition  $\phi$ . For a phase Ph,  $\phi$  represents a condition that depends solely on intrinsic properties of Ph. For instance, one might say that if John is a Living Person then he is a Person who has the property of being alive or, if Spot is a Puppy then it is a Dog who has the property of being less than one year old. For a role Rl, conversely,  $\phi$  depends on extrinsic (relational) properties of Rl. For example, one might say that if John is a Student then John is a Person

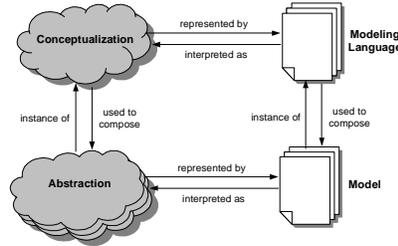
---

<sup>1</sup>Red is used here as an object universal whose instances are particulars like a red apple  $x$ , not as a quality universal whose instances are particulars such as the specific redness of  $x$  [1].



latter *domain abstractions*. Take as an example the domain of genealogical relations in reality. A certain conceptualization of this domain can be constructed by considering concepts such as *Person, Man, Woman, Father, Mother, Offspring, being the father of, being the mother of*, among others. By using these concepts, we can articulate a domain abstraction (i.e., a mental model) of certain facts in reality such as, for instance, that *a man named John is the father of another man named Paul*.

Conceptualizations and Abstractions are immaterial entities that only exist in the mind of the user or a community of users of a language. In order to be documented, communicated and analyzed they must be captured, i.e. represented in terms of some concrete artifact. This implies that a language is necessary for representing them in a concise, complete and unambiguous way. Figure 3 depicts the distinction between an abstraction and its representation, and their relationship with conceptualization and representation language. In the scope of this work the representation of a domain abstraction in terms of a representation language  $\mathcal{L}$  is called a *model* and the language  $\mathcal{L}$  used for its creation is called a *modeling language*.

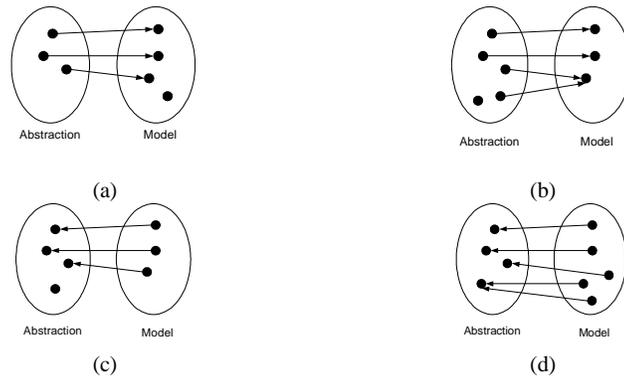


**Fig. 3.** Relations between Conceptualization, Abstraction, Modeling Language and Model

In order for a model  $\mathcal{M}$  to faithfully represent an abstraction  $\mathcal{A}$ , the modeling primitives of the language  $\mathcal{L}$  used to produce  $\mathcal{M}$  should faithfully represent the domain conceptualization  $\mathcal{C}$  used to articulate the represented abstraction  $\mathcal{A}$ . The *Domain Appropriateness* of a language is a measure of the suitability of a language to model phenomena in a given domain, or in other words, of its truthfulness to a given domain in reality. On a different aspect, different languages and specifications have different measures of pragmatic adequacy [1]. *Comprehensibility appropriateness* refers to how easy is for a user a given language to recognize what that language's constructs mean in terms of domain concepts and, how easy is to understand, communicate and reason with the specifications produced in that language. These two quality criteria can be systematically evaluated by comparing, on one hand, a concrete representation of the worldview underlying that language (captured by that language's *metamodel*) to, on the other hand, a concrete representation of a domain conceptualization, or a *domain ontology*. The truthfulness to reality (*domain appropriateness*) and conceptual clarity (*comprehensibility appropriateness*) of a modeling language depend on the level of homomorphism between these two entities [1]. The stronger the match between an abstraction in reality and its representing model, the easier is to communicate and reason with that model.

The mapping from concepts-to-constructs and its inverse (i.e., constructs-to-concept) are named here a *representation* and *interpretation* mappings, respectively. In [1], we discuss a number of properties that should be reinforced for isomorphic mappings to take place between an ontology  $O$  representing a domain  $\mathcal{D}$  and a language's metamodel. If isomorphism can be guaranteed, the implication for the human agent who interprets a diagram (model) is that his interpretation correlates precisely and uniquely with an abstraction being represented. By contrast, where the correlation is not an isomorphism then there may potentially be a number of unintended abstractions which would match the interpretation. These properties are briefly discussed in the sequel and are illustrated in figure 4: (a) Soundness: A language  $\mathcal{L}$  is *sound* w.r.t. to a domain  $\mathcal{D}$  iff every modeling primitive in the language has an interpretation in terms of a domain concept in the ontology  $O$ ; (b) Completeness: A language  $\mathcal{L}$  is *complete* w.r.t. to a domain  $\mathcal{D}$  iff every concept in the ontology  $O$  of that domain is represented in a modeling primitive of that language; (c) Lucidity: A language  $\mathcal{L}$  is *lucid* w.r.t. to a domain  $\mathcal{D}$  iff every modeling primitive in the language represents at most one domain concept in  $O$ . (d) Laconicity: A language  $\mathcal{L}$  is *laconic* w.r.t. to a domain  $\mathcal{D}$  iff every concept in the ontology  $O$  of that domain is represented at most once in the metamodel of that language. In [1], we also provide a methodological framework for systematically assessing these properties given a language and a domain.

*Unsoundness, Non-Lucidity, Non-Laconicity and Incompleteness* violate what the philosopher of language H.P.Grice [11] names *conversational maxims* that states that a speaker is assumed to make contributions in a dialogue which are *relevant, clear, unambiguous, and brief, not overly informative and true according to the speaker's knowledge*. Whenever models do not adhere to these conversational maxims, they can communicate incorrect information and induce the user to make incorrect inferences about the semantics of the domain.



**Fig. 4.** Examples of Lucid (a) and Sound (b) representational mappings from Abstraction to Model; Examples of Laconic (c) and Complete (d) interpretation mappings from Model to Abstraction.

In regards to the property of completeness, when mapping the elements of a domain ontology to a language metamodel we must guarantee that these elements are repre-

sented in their full formal descriptions. In other words, the metamodel  $\mathcal{MT}$  of language  $\mathcal{L}$  representing the domain ontology  $O$  must also represent this ontology's full axiomatization. In formal, model-theoretic terms, this means that these entities should have the same set of logical models. In [1], we discuss this topic in depth and present a formal treatment of the idea. The set of logical models of  $O$  represent the state of affairs in reality deemed possible by a given domain conceptualization. In contrast, the set of logical models of  $\mathcal{MT}$  stand for the world structures which can be represented by the grammatically correct specifications of language  $\mathcal{L}$ . In summary, we can state that if a domain ontology  $O$  is fully represented in a language metamodel  $\mathcal{MT}$  of  $\mathcal{L}$ , then the only grammatically correct models of  $\mathcal{L}$  are those which represent state of affairs in reality deemed possible by the domain conceptualization represented by  $O$  (termed *intended world structures*).

In the beginning of this section, we have exemplified the notions discussed above by referring to the domain of genealogical relations. This exemplifies what is named a *material domain* in the literature. Accordingly, a modeling language designed to represent phenomena in this domain is named a *Domain-Specific Modeling Language*. However, take the case of a (domain-independent) *general conceptual modeling language* (e.g., EER, ORM, UML). What should be real-world conceptualization that this language should commit to? The position defended here is that it should be a system of general categories and their ties, which can be used to articulate domain-specific common sense theories of reality. This meta-conceptualization should comprise a number of domain-independent theories (e.g., types and instantiation, taxonomic structures, identity, existential dependence, etc.) which are able to characterize aspects of real-world entities irrespective of their particular nature. The development of such general theories of reality is the business of the philosophical discipline of *Formal Ontology* in philosophy and a concrete artifact representing one of these meta-conceptualizations is a *Foundational Ontology*. An example of a Foundational Ontology is the UFO Ontology presented in the section 2.

## 4 Evaluating and Re-Designing the UML 2.0 Metamodel

In the sequel we start by constructing representation and interpretation mappings between the concrete metaclasses of the UML metamodel presented in [12] and the ontological categories comprising the foundational ontology depicted in figure 2.

**Class and Generalization:** We begin by focusing on a special sense of the UML metaclass *Class* (see figure 5). By class hereby we mean the notion of a first-order class, as opposed to *powertypes*, and one whose instances are single objects, as opposed to *association classes*, whose instances are tuples of objects. In this sense, if we make a representation mapping from UFO to the UML metamodel, we can map the category of *Monadic Universal* to the UML element of a *Class*. However, by carrying on this process, we realize that in UML there are no modeling constructs that represent the ontological categories specializing *Object Universal* in figure 2. In other words, there are ontological concepts prescribed by our reference ontology that are not represented by any modeling construct in the language. This amounts to a case of

***incompleteness***. Moreover, as discussed in section 2.4, the theory of object universals comprising UFO prescribes a number of constraints governing the relations between these different types of universals. By not taking this into account, the UML meta-model admits a number of grammatically correct specifications and logical models which are not representations of valid state of affairs (intended world structures) according to the reference ontology.

**Attributes, Attribute Values and Datatypes:** If we now consider the other types of monadic universals accounted in our theory, namely, *moment* and *relator universals* we shall realize that they too lack representation in the language metamodel. This amounts to another case of ***incompleteness*** in the modeling language.

As discussed at length in [1], quality universals are typically not represented in a conceptual model explicitly but via *attribute functions* that map each of their instances to points in a quality structure. For example, suppose we have the universal Apple (an object universal) whose instances exemplify the universal Weight. We say in this case that the quality universal Weight *characterizes* the kind Apple. Thus, for an arbitrary instance  $x$  of Apple there is a quality  $w$  (instance of the quality universal Weight) that inheres in  $x$ . Associated with the universal Weight, and in the context of a given measurement system (e.g., the human perceptual system), there is a quality dimension *weightValue*, which is a set isomorphic to the half line of positive integers obeying the same ordering structure. Quality structures are taken here to be theoretical abstract entities modeled as sets. In this case, we can define an *attribute function* (another abstract theoretical entity) *weight(Kg)*, which maps each instance of apple (and in particular  $x$ ) onto a point in a quality dimension, i.e., its *quale*. Thus, attribute functions are the ontological interpretation of UML attributes, i.e., UML Properties which are owned by a given classifier (figure 5).

As any property, a UML attribute is a typed element and, thus, it is associated to Type. Type constrains the sort of entities that can be assigned to slots representing that attribute in instances of their owning classifier. Since Classifier is a specialization of Type, we have that both Classes and Datatypes can be the associated types of an UML attribute. In other words, an attribute represents both an attribute function and a sort of a *relational image function*<sup>2</sup> that, for example, in binary relation *ownership* between the classifiers Person and Car, maps a particular Person  $p$  to all instances of Car that are associated with  $p$  via this relation (i.e., all cars owned by  $p$ ). From a software design and implementation point of view, an attribute represents a method implemented by the owning class, and the type of the attribute represents the returning type of that method. However, from a conceptual point of view, in the UML metamodel an attribute stands both for a monadic (intrinsic) and for a relational property and, thus, it can be considered a case of ***non-lucidity***. On another perspective, UML offers an alternative notation for the representation of attributes, namely, *navigable end names*. That is, the same ontological concept (*attribute function*) is represented in the language via more than one construct, which characterizes a case of ***non-laconicity***.

---

<sup>2</sup>A relational image function is formally defined as follows: Let  $R$  be a binary relation defined for the two sets  $X$  and  $Y$ . The function  $\mathbf{Im}$  with signature  $\mathbf{Im}(\_,\_): X \times (X \Leftrightarrow Y) \rightarrow \wp(Y)$  is defined as  $\mathbf{Im}(x, R) = \{y \mid (x, y) \in R\}$ .

The DataType associated with an attribute A of class C is the representation of the quality structure that is the co-domain of the attribute function represented by A. In other words, a quality structure is the ontological interpretation of the UML DataType construct. Moreover, we have that a multidimensional quality structure (quality domain) is the ontological interpretation of the so-called *Structured DataTypes*. Quality domains are composed of multiple integral dimensions. This means that the value of one dimension cannot be represented without representing the values of others. The fields of a datatype representing a quality domain QD represent each of its integral quality dimensions. Alternatively we can say that each field of a datatype should always be interpreted as representing one of the integral dimensions of the QD represented by the datatype. The constructor method of the datatype representing a quality domain must reinforce that its tuples always have values for all the integral dimensions. Finally, an algebra can be defined for a DataType so that the relations constraining and informing the geometry of represented quality dimensions are also suitably characterized. As discussed in [1], according to the UML specification, a DataType is an abstract entity that collects other abstract entities (“*pure values*”) that can be multiply referred, i.e., a DataType is not a multiply instantiated universal but an abstract particular (set) with other particulars as members.

**Associations:** In UML, the association meta-construct is used to represent the ontological concept of *Relation*. Relations in UFO can be *material* or *formal*. The latter in turn can be subdivided in *basic formal relations* (internal relations) and *comparative formal relations*. Since class diagrams only represent universals, the only basic formal relations among the ones we have considered that should have a representation in these models are the relations of *characterization*, *mediation* and *derivation*. These concepts have no representation in the UML metamodel, which characterizes another case of *incompleteness*.

The association class construct in UML exemplifies a case of *non-lucidity*, since “an association class can have as instances either (a) a n-tuple of entities which classifiers are endpoints of the association; (b) a n+1-tuple containing the entities which classifiers are endpoints of the association plus an instance of the objectified association itself” [13]. This is to say that an association class can be interpreted both as a *relation* and what is termed in the literature a *factual universal* [1]. In short, if the relator r connects (mediates) the entities  $a_1, \dots, a_n$  then this yields a new particular that is denoted by  $\langle r: a_1, \dots, a_n \rangle$ . Particulars of this latter sort are called *material facts*.

In addition to that, since the “*instance of the objectified association itself*” is supposed to be an object identifier for the n-tuple, one cannot represent cases in which the same relator mediates multiple occurrences of the same n-tuple. As an example of the latter suppose the following situation. Suppose a Treatment relator universal and a TreatedIn material relation (derived from it) defined between Patients and Medical Units. Now suppose that treatment  $t_1$  mediates the individuals John, and the medical units  $MedUnit_{\#1}$  and  $MedUnit_{\#2}$ . In this case, we have as instances of Treatment both facts  $\langle t_1: John, MedUnit_{\#1} \rangle$  and  $\langle t_1: John, MedUnit_{\#2} \rangle$ . However, this cannot be represented in such a manner in UML. In UML,  $t_1$  is supposed to function as an object identifier for a unique tuple. Thus, if the fact  $\langle t_1: John, MedUnit_{\#1} \rangle$  holds then  $\langle t_1: John, MedUnit_{\#2} \rangle$  does not, or alternatively, John and  $MedUnit_{\#2}$  must be mediated by

another relator. These are, nonetheless, unsatisfactory solutions, since it is the very same relator Treatment that connects one patient to a number of different medical units. In conclusion, association classes on one hand represent a case of non-lucidity, on the other hand, allow for a case of construct incompleteness at the instance level.

**Interfaces:** According to the UML specification, an interface is a declaration of a coherent set of features and obligations. It can be seen as a kind of contract that partition and characterize groups of properties which must be fulfilled by any instance of a classifier that implements that interface. In an interpretation mapping from the UML metamodel to UFO, an interface qualifies as a case of *unsoundness*. This means that, being merely a design and implementation construct, there is no category in the conceptual modeling ontology proposed here that serve as the ontological interpretation for a UML interface.

## 5 Reinforcing the Isomorphism between UFO and UML

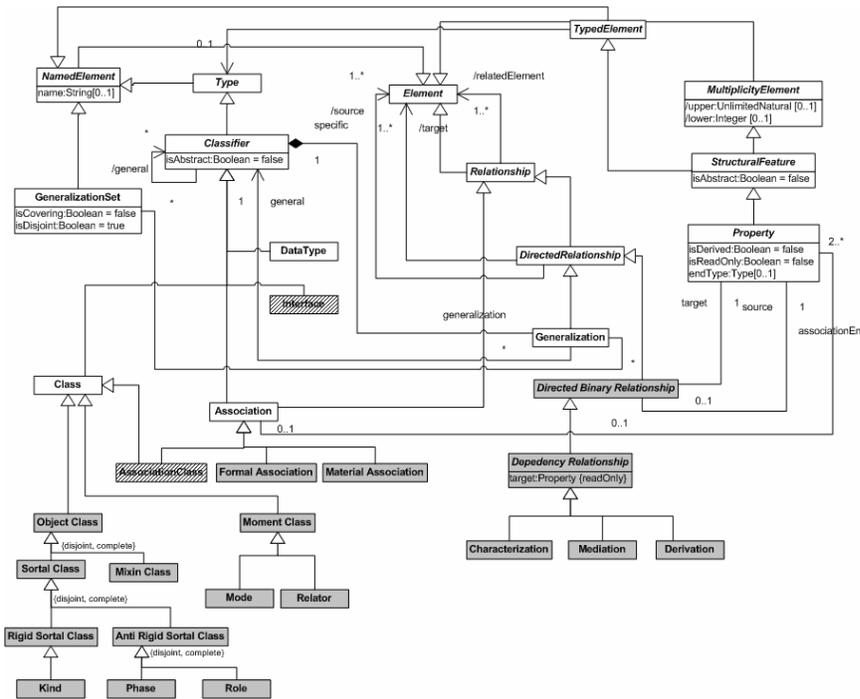


Fig 5. The Redesigned UML 2.0 metamodel

As demonstrated in the previous section, from an ontological point of view, UML includes cases of ontological incompleteness, unsoundness, non-lucidity and non-

laconicity. In the sequel, we discuss briefly how these problems have been solved to produce an ontologically well-founded version of UML for conceptual modeling [1].

**Incompleteness:** In order to remedy this problem, we propose extensions to the UML metaclass *Class* that represent different types of monadic universals. As shown in figure 5, these extensions represent finer-grained distinctions between different sorts of *object types* as well as the notions of *mode* and *relator* universals and *material* and *formal* relations.

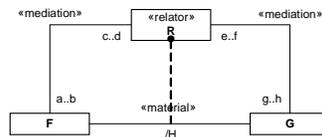
Another example of incompleteness identified in the previous section is w.r.t. the representation of different types of basic formal relations, namely, the relations of *characterization*, *mediation* and *derivation*. There are a number of common characteristics shared by these relations. Firstly, they are all *directed relations*. In the case of characterization, the source is a class representing a mode universal; in the case mediation, one representing a relator universal; in the case of derivation, a material relation. In the first two cases, the target is a class representing either an object or moment universal, while in the case of derivation, the target is necessarily one representing a relator universal. Secondly, all these relations are mapped in the instance level to an *existential dependency* relation between the corresponding source particulars and their depended particulars. This has the following consequences in the metamodel: (i) the association end connected to the target class must have the cardinality constraints of one and exactly one, since every moment or fact is a dependent entity; (ii) the association end connected to the target class must have the meta-attribute *isReadOnly = true*, since existential dependency is modally constant; (iii) existential dependency relations are always binary relations.

In order to account to all these requirements, we extend the original UML metamodel by extending the metaclass direct relationship with the metaclasses direct binary relationship, dependency relationship, and finally, the basic formal relations of characterization, mediation and derivation (figure 5). Finally, since a relator is dependent (mediates) on at least two numerically distinct entities, we have the following additional constraint: (iv) Let  $R$  be a class representing a relator universal and let  $\{C_1 \dots C_2\}$  be a set of classes mediated by  $R$  (related to  $R$  via a *mediation* relation). Finally, let  $lower_{C_i}$  be the value of the minimum cardinality constraint of the association end connected to  $C_i$  in the mediation relation. Then,  $(\sum_{i=1}^n lower_{C_i}) \geq 2$ .

Asides from incorporating metaclasses that represent the missing ontological concepts, the extended UML metamodel must also include a number of constraints derived from the constraints in the ontology that restrict the ways the introduced elements can be related (see constraints on section 2.4 as well as (i-iv) above). The goal is to have a metamodel such that all grammatically correct specifications according to this metamodel have logical models that are *intended world structures* of the conceptualizations they are supposed to represent. In [1], besides from extending the UML metamodel in order to represent the ontological concepts discussed above, we define a *profile* that implements the metaclasses of this (extended) UML metamodel, as well as their interrelationships and constraints. By using this profile, for example, the concrete *object classes* in figure 5 are represented in conceptual models as stereotyped classes representing each of the considered ontological distinctions. Likewise, the

admissible relations between these ontological categories, derived from the postulates of our theory, are represented in the profile as syntactical constraints governing the admissible relations between the corresponding stereotyped classes. A fragment of this profile is shown in table 1. For the complete definition of this profile as well as an in depth discussion motivating its elements one should refer to [1].

**Non-Lucidity:** As discussed in the previous section, in UML, attributes represent both the ontological concepts of attribute functions and relational image functions, which is case of non-lucidity. To eliminate this problem, we prescribe that attributes should only be used to represent attribute functions. As consequence, their associated types should be restricted to DataTypes only. The UML construct of association classes amounts to a case of both non-lucidity (since it represents a factual and relator universal) and incompleteness (since one cannot represent cases in which the same relator mediates multiple occurrences of the same n-tuple). We propose, therefore, to disallow the use of association classes in UML for the purpose of conceptual modeling. In contrast, we propose to represent relational properties explicitly. We use the stereotype «relator» to represent the ontological category of relator universals. Relator universals can induce material relations. A material relation induced by a relator universal  $R$  is represented by a UML association stereotyped as «material» (UML base class *association*). The basic formal relation *derivation* is represented by a dashed line with a black circle in one of the ends (see figure 6). A derivation relation is a specialized type of relationship between the stereotyped association representing the derived «material» association and the stereotyped class representing the founding «relator» universal. The black circle represents the role of foundation of the relator universal side. Every «material» association must be the association end of exactly one derivation relation. Still on figure 6, from the cardinality constraints of the two «mediation» relations we can derive the maximum cardinality of the derivation relation (on the material relation end) and the cardinality constrains on both association ends of the material relation itself. For instance, the upper constraint  $\delta$  on the end connected to  $G$  in the  $H$  relation is the result of  $(d \times h)$ ; the upper constraint  $\beta$  in the end connected to  $F$  is the result of  $(f \times b)$ . The upper constraint  $\phi$  in the end  $H$  of the derivation relation is the result of  $(b \times h)$ . Likewise, we can calculate the derived minimum cardinality constraints in the following manner:  $\gamma = c \times g$ ;  $\alpha = e \times a$ , and  $\varepsilon = a \times g$ . We should highlight that the relator particular is the actual instantiation of the corresponding relational property (the objectified relation). Material relations stand merely for the facts derived from the relator particular and its mediating entities. Therefore, we claim that the representation of the relators of material relations must have primacy over the representation of the material relations themselves. In other words, the representation of «material» relations can be omitted but whenever a «material» is represented it must be connected to an association end of a derivation relation.



**Fig 6.** Representing Material Relations and their founding Relators.

Finally, we use the stereotype «formal» to represent comparative formal relations. Comparative formal relations and material relations are derived relations. Whilst the former are derived from intrinsic properties of the related entities, the latter are derived from relators and their mediating entities. Therefore, we prescribe that UML associations stereotyped as «material» must have the meta-attribute (isDerived = true). *Mutatis Mutandis*, we use the same meta-attribute to represent formal relations which are not *internal relations*, i.e., which are comparative.

**Non-Laconicity:** In the UML notation, the same ontological concept of attribute functions has two representations in terms of the language constructs, namely, the textual notation for attributes and navigable association ends. This situation could be justified from a pragmatic point of view if navigable ends were used to model only *structured DataTypes*, and if the textual notation for attributes were only used to model the simple ones. However, in the current UML metamodel, there is no constraint on using both notations for both purposes. To eliminate the potential ambiguity of this situation, we propose to use navigable ends to represent only attribute functions whose co-domains are *multidimensional quality structures* (quality domains). Conversely, those functions whose co-domains are quality dimensions should only be represented by the attributes textual notation.

**Unsoundness:** An example of a UML construct which lacks an ontological interpretation is the construct of *Interfaces*. For this reason, we propose that the use of this construct should be disallowed in an ontologically well-founded version of UML. In figure 5, the metaclasses interface and association classes which have been disallowed in this metamodel according to our analysis appear as hachured classes.

## 6 Final Considerations

The development of a well-grounded, axiomatized upper level ontology is an important step towards the definition of real-world semantics for conceptual modeling diagrammatic languages. In this paper, we use present the ontology **UFO (Unified Foundational Ontology)**, which has been designed with the specific purpose of serving as a foundational theory for conceptual modeling. Additionally, we briefly present an ontology-based framework for evaluating the *domain* and *comprehensibility appropriateness* of modeling languages. The framework defines a systematic method for comparing the *metamodel* of a language with a concrete representation of a conceptualization of a given subject domain, termed a *reference ontology*. The paper illustrates the usefulness of the UFO ontology as a reference ontology and application of the method by evaluating and redesigning the UML metamodel for the purpose of conceptual modeling.

In [1], the re-designed UML metamodel discussed here has been used in the implementation of a UML profile for Conceptual Modeling. The profile comprises of: (i) a set of stereotypes representing ontological distinctions proposed by the theory (ii) constraints on the possible relations to be established between these elements, representing the postulates of the theory. By using this profile, we were able to propose a

number of sound engineering tools and principles, and methodological guidelines for the practice of conceptual modeling such as the *role modeling design pattern* and the *visual patterns for delimiting the scope of transitive parthood relations*, both presented in [1].

Finally, it is important to emphasize that, in this article, only a fragment of UFO is presented. In particular, a fragment of the Ontology of Endurants in UFO named UFO-A. In [2,16], UFO is presented in three compliance sets, namely, UFO-A: an *Ontology of Endurants*; UFO-B: an *Ontology of Perdurants*, and UFO-C, which is built upon UFO-A and B to compose an *Ontology of Social Concepts*. Although UFO-B and C do not enjoy the same level of maturity and stability as UFO-A, they have been recently employed with success in the analysis of other conceptual modeling languages and frameworks such as REA (Resource-Event Action) [2], Tropos and AORML [14], EM-BRACE [15] and the ODE Software Process Ontology [16].

**Table 1.** Fragment of the UML profile implementing the metamodel of figure 5.

Metaclass	Description
«kind» A	<i>Kind</i> represents rigid, relationally independent object universals that supply a principle of identity for their instances.
<b>Constraints</b>	
<ol style="list-style-type: none"> <li>1. Every object represented in a conceptual model using this profile must be an instance of a kind, directly or indirectly. This means that every concrete element of this profile used in a class diagram (isAbstract = false) must include in its general collection one class stereotyped as «kind»;</li> <li>2. An object represented in a conceptual model using this profile cannot be an instance of more than one kind. This means that any stereotyped class in this profile used in a class diagram must not include in its general collection more than one kind. Moreover, a kind must also not include another kind in its general collection;</li> <li>3. A Class representing a rigid object universal cannot be a subclass of a Class representing an anti-rigid universal. Thus, a kind cannot have as a supertype (must not include in its general collection) a member of {«phase», «role»}.</li> </ol>	
Metaclass	Description
Derivation Relation 	Dependency Relationship A derivation relation represents the formal relation of derivation that takes place between a material relation and the relator universal this material relation is derived from.
<b>Constraints</b>	
<ol style="list-style-type: none"> <li>1. A derivation relation must have one of its association ends connected to a relator universal (the black circle end) and the other one connected to a material relation (self.target.oclsTypeOf(Relator)=true, self.source.oclsTypeOf(Material Association)=true);</li> <li>2. derivation associations are always binary associations;</li> <li>3. The black circle end of the derivation relation must have the cardinality constraints of one and exactly one (self.target.lower = 1 and self.target.upper = 1);</li> <li>4. The black circle end of the derivation relation must have the property (self.target.isreadOnly = true);</li> <li>5. The cardinality constraints of the association end connected to the material relation in a derivation relation are a product of the cardinality constraints of the «mediation» relations of the relator universal that this material relation derives from. This is done in the manner previously shown in this section. However, since «mediation» relations require a minimum cardinality of one on both of its association ends, then the minimum cardinality on the material relation end of a derivation relation must also be ≥ 1 (self.source.lower ≥ 1).</li> </ol>	

## References

1. Guizzardi, G. 'Ontological Foundations for Structural Conceptual Models', PhD Thesis, University of Twente, The Netherlands, 2005.
2. Guizzardi, G.; Wagner, G. On a Unified Foundational Ontology and some Applications of it in Business Modeling, *Ontologies and Business Systems Analysis*, Michael Rosemann and Peter Green (Eds.), IDEA Publisher, 2005.
3. Gärdenfors, P. 'Conceptual Spaces: the Geometry of Thought', MIT Press, USA, 2000.
4. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A. 'Ontology Library', WonderWeb Deliverable D18, 2003.
5. Heller, B., Herre, H. Ontological Categories in GOL. *Axiomathes* 14: 71-90, Kluwer Academic Publishers, 2004.
6. Smith, B.; Mulligan, K., A Relational Theory of the Act, *Topoi* (5/2), 115-30, 1986.
7. Schneider, L. Formalised Elementary Formal Ontology, ISIB-CNR Technical Report 03/2002, [online: <http://www.loa-cnr.it/Publications.html>], 2002.
8. Lowe, E.J., *The Four Category Ontology*, Oxford University Press, 2006.
9. Ferrario, R.; Oltramari, A. Towards a Computational Ontology of the Mind, *Proceedings of the 3rd International Conference on Formal Ontology in Information Systems (FOIS)*, Torino, Italy, 2004.
10. Masolo, C.; Guizzardi, G.; Vieu, L.; Bottazzi, E.; Ferrario, R. (2005): *Relational Roles and Qua Individuals*, AAAI Fall Symposium on Roles, an Interdisciplinary Perspective, Virginia, USA.
11. Grice, H.P. 'Logic and conversation', In: *Syntax and Semantics: Vol 3, Speech Acts* (P. Cole & J. Morgan, eds). Academic Press, New York, pp. 43-58, 1975.
12. Object Management Group, UML 2.0 Superstructure Specification, Doc.# ptc/03-08-02, Aug. 2003.
13. Breu, R.; Hinkel, U.; Hofmann, C.; Klein, C.; Paech, B.; Rumpe, B.; Thurner, V. 'Towards a Formalization of the Unified Modeling Language', *Proceedings fo the 11th ECOOP*, Jyväskylä, Finland, 1997.
14. Guizzardi, R.S.S.; Guizzardi, G. "Integrating Agent-Oriented Modeling Languages Using a Foundational Ontology", in *Social Modeling for Requirements Engineering*, P. Giorgini, N. Maiden, J. Mylopoulos, E. Yu (eds.), *Cooperative Information Systems Series*, MIT Press, 2008 (forthcoming).
15. Goedertier, S.; Haesen, R.; Vanthienen, J., An ontology for rule-based business process modeling, *International Journal of Business Process Integration and Management (IJBPIIM)*, Inderscience Publishers, 2008 (forthcoming).
16. Guizzardi, R.S.S.; Guizzardi, G. "Integrating Agent-Oriented Modeling Languages Using a Foundational Ontology", in *Social Modeling for Requirements Engineering*, P. Giorgini, N. Maiden, J. Mylopoulos, E. Yu (eds.), *Cooperative Information Systems Series*, MIT Press, 2008 (forthcoming).
17. Guizzardi, G.; Vasconcelos, J.O.; Segrini, B.; Falbo, R.; Guizzardi, R.S.S. "Grounding Software Domain Ontologies in the Unified Foundational Ontology(UFO): The case of the ODE Software Process Ontology", *11<sup>th</sup> Iberoamerican Workshop on Requirements Engineering and Software Environments*, Recife, Brazil, 2008.