On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models

Giancarlo Guizzardi
Federal University of Espírito Santo (UFES),Vitória, Brazil
Laboratory for Applied Ontology (ISTC-CNR),Trento, Italy
guizzardi@loa-cnr.it

Abstract. In philosophy, the term ontology has been used since the 17th century to refer both to a philosophical discipline (Ontology with a capital "O"), and as a domain-independent system of categories that can be used in the conceptualization of domain-specific scientific theories. In the past decades there has been a growing interest in the subject of ontology in computer and information sciences. In the last few years, this interest has expanded considerably in the context of the Semantic Web and MDA (Model-Driven Architecture) research efforts, and due to the role ontologies are perceived to play in these initiatives. In this paper, we explore the relations between Ontology and ontologies in the philosophical sense with domain ontologies in computer science. Moreover, we elaborate on formal characterizations for the notions of ontology, conceptualization and metamodel, as well as on the relations between these notions. Additionally, we discuss a set of criteria that a modeling language should meet in order to be considered a suitable language to model phenomena in a given domain, and present a systematic framework for language evaluation and design. Furthermore, we argue for the importance of ontology in both philosophical senses aforementioned for designing and evaluating a suitable general ontology representation language, and we address the question whether the so-called Ontology Web languages can be considered as suitable general ontology representation languages. Finally, we motivate the need for two complementary classes of modeling languages in Ontology Engineering addressing two separate sets of concerns.

Keywords: Formal Ontology, Conceptual Modeling, Metamodeling, Language Adequacy

1 Introduction

The Webster dictionary [1] defines the term ontology as:

(D1). a branch of metaphysics concerned with the nature and relations of being;
(D2). a particular theory about the nature of being or the kinds of existents;
(D3). a theory concerning the kinds of entities and specifically the kinds of abstract entities that are to be admitted to a language system.

Etymologically, *ont-* comes from the present participle of the Greek verb *einai* (to be) and, thus, the latin word *Ontologia* (*ont-* + *logia*) can be translated as *the study of existence* The term was coined in the 17th century in parallel by the philosophers Rudolf Göckel in his *Lexicon philosophicum* and by Jacob Lorhard in his *Ogdoas Scholastica*, but popularized in philosophical circles only in 18th century with the publication in 1730 of the *Philosophia prima sive Ontologia* by the German philosopher Christian Wolff.

In the sense (D1) above, ontology is the most fundamental branch of metaphysics. Aristotle was the first western philosopher to study metaphysics systematically and to lie out a rigorous account of ontology. He described (in his *Metaphysics* and *Categories*) ontology as *the science of being qua being*. According to this view, the business of ontology is to study the most general features of reality and real objects, i.e., the study of the generic traits of every mode of being. As opposed to the several specific scientific disciplines (e.g., physics, chemistry, biology), which deal only with entities that fall within their respective domain, ontology deals with transcategorical relations, including those relations holding between entities belonging to distinct domains of science, and also by entities recognized by *common sense.*

In the beginning of the 20th century the German philosopher Edmund Husserl coined the term *Formal Ontology* as an analogy to Formal Logic. Whilst Formal Logic deals with formal logical structures (e.g., truth, validity, consistency) independently of their veracity, Formal Ontology deals with formal ontological structures (e.g., theory of parts, theory of wholes, types and instantiation, identity, dependence, unity), i.e., with formal aspects of objects irrespective of their particular nature. The unfolding of Formal Ontology as a philosophical discipline aims at developing a system of general categories and their ties, which can be used in the development of scientific theories and domain-specific common sense theories of reality. In other words, Ontology (as a discipline, with capital O) in the first sense of Webster's definition aforementioned contributes to the development of ontologies in the second sense. The first ontology developed in sense (D2) is the set of theories of Substance and Accidents developed by Aristotle in his *Methaphysics* and *Categories*. Since then, ontological theories have been proposed by innumerous authors in philosophy, and more recently also in the area of *Applied Ontology* in computer science (e.g., DOLCE, GFO, OCCHRE, UFO).

The term *"ontology"* in the computer and information science literature appeared for the first time in 1967, in a work on the foundations of data modeling by S. H. Mealy, in a passage where he distinguishes three distinct realms in the field of data processing, namely: (i) *"the real world itself"*; (ii) *"ideas about it existing in the minds of men"*; (iii) "*symbols on paper or some other storage medium"*. Mealy concludes the passage arguing about the existence of things in the world regardless of their (possibly) multiple representations and claiming that *"This is an issue of ontology, or the question of what exists"* [2,p.525]. In the end of this passage, Mealy includes a reference to Quine's essay *"On What There Is"* [3]. In an independent manner, yet another sub-field of computer science, namely Artificial Intelligence (AI) began to make use of what came to be known as *domain ontologies*. Since the first time the term was used in AI by Hayes [4] and since the development of his naïve physics ontology of liquids [5], a large amount of domain ontologies have been developed in a multitude of subject areas. In the past five years, an explosion of works related to ontology has happened in computer science, chiefly motivated by the growing interest on the Semantic Web, and by the key role played by them in that initiative.

An important point that should be emphasized is the difference in the senses of the term used by the information systems, on one side, and artificial intelligence and semantic web communities on the other. In information systems, the term ontology has been used in ways that conform to its definitions in philosophy (in both senses D1 and D2). As a system of categories, an ontology is independent of language: Aristotle's ontology is the same whether it is represented in English, Greek or First-Order Logic. In contrast, in most of other areas of computer science (the two latter areas included), the term ontology is, in general, used as a concrete engineering artifact designed for a specific purpose, and represented in a specific language.

In the light of these contrasting notions of ontologies, a number of question begging issues become manifest: What exactly is a domain ontology? How does it relate to other concrete representations such as conceptual models and metamodels? How does it related to ontology in the philosophical senses (D1-D3) aforementioned? Additionally, during the years many languages have been used to represent domain ontologies. Examples include Predicate Calculus, KIF, Ontolingua, UML, EER, LINGO, ORM, CML, DAML+OIL, F-Logic, OWL. What are the characteristics that a suitable language to represent conceptual models, in general, and domain ontologies, in particular should have? In particular, are the semantic web languages suitable ontology representation languages?

The objective of this article is to offer answers to these questions. In the next section, we start by discussing the relation between reality, conceptualization and language, and by briefly introducing a framework that can be used to systematically evaluate and re-design a modeling language w.r.t. its suitability to model phenomena in a given domain. In section 3, we elaborate on the notion of a language metamodel. In section 4, we provide a formal account for the notion of domain ontology as well as for its relation to conceptualization and language metamodel as discussed in section 2. In section 5, we advocate the need for an ontologically well-founded system of categories that should underlie a suitable ontology representation language (i.e., an ontology in the sense D2 and D3), and discuss the role played by Formal Ontology in Philosophy (Ontology in the sense D1) in the development of such a system. In section 6, we make use of the framework of section 2 and provide a concrete example to illustrate many of the notions discussed in the article, namely, those of foundational and domain ontology, ontology representation language, domain-specific language, and (meta)model. In section 7, we motivate the need for two complementary classes of representation languages in Ontology Engineering: one class populated by philosophically well-founded languages, focused on expressivity and conceptual clarity, and another one populated by languages focused on computation-oriented concerns (e.g., decidability, efficient automated reasoning, etc.). In section 8, we conclude this article with a summary of the most important points discussed herein.

2 Conceptualization and Language

One of the main success factors behind the use of a modeling language lies in the language's ability to provide to its target users a set of modeling primitives that can directly express relevant domain concepts, comprising what we name here a *domain conceptualization*. The elements constituting a *conceptualization* of a given domain are used to articulate abstractions of certain state of affairs in reality. We name the 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 1 below represents the relation between a language, a conceptualization and the portion of reality that this conceptualization abstracts. This picture depicts the well-known *Ullmann's triangle* [6]. This triangle derives from that of Ogden and Richards [7] and from Ferdinand de Saussure [8], on whose theories practically the whole modern science of language is based.

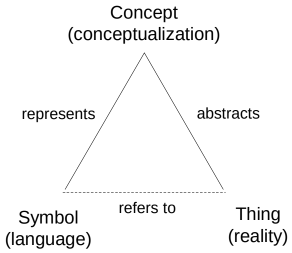

Fig.1. Ullmann's Triangle: the relations between a thing in reality, its conceptualization and a symbolic representation of this conceptualization

The *represents* relation concerns the definition of language *$\mathcal{L}$'s real-world semantics*. The dotted line between language and reality in this figure highlights the fact that the relation between language and reality is always intermediated by a certain conceptualization [9]. This relation is elaborated in Figure 2 that 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 specification* (or simply *model*, *specification*, or *representation*) and the language $\mathcal{L}$ used for its creation is called a *modeling* (or *specification*) *language*.

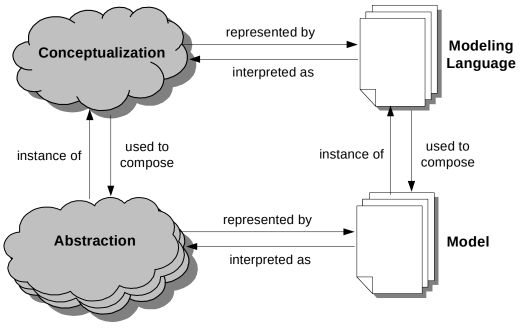

Fig. 2. Relations between Conceptualization, Abstraction, Modeling Language and Model

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 of a language to a given domain in reality. On a different aspect, different languages and specifications have different measures of pragmatic adequacy [10]. *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. The measures of these two quality criteria for a given language and domain are aspects of the *represents* relation depicted in figure 1, and they 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. The stronger the match between an abstraction in reality and its representing model, the easier is to communicate and reason with that model.

In [10], we discuss a number of properties that should be reinforced for an isomorphic mapping to take place between an ontology $\mathcal{O}$ representing a domain $\mathcal{D}$ and a domain 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 3: (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 $\mathcal{O}$; (b) *Completeness:* A language $\mathcal{L}$ is *complete* w.r.t. to a domain $\mathcal{D}$ iff every concept in the ontology $\mathcal{O}$ 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 $\mathcal{O}$. (d) *Laconicity:* A language $\mathcal{L}$ is *laconic* w.r.t. to a domain $\mathcal{D}$ iff every concept in the ontology $\mathcal{O}$ of that domain is represented at most once in the metamodel of that language. In the same article, we also provide a methodological framework for assessing these properties given a language and a domain. Such framework has been applied in a number of case studies. The most comprehensive example being [10] with the evaluation and re-design of UML for the purpose of conceptual modeling, but also [11], in which this framework is employed to design an agent-oriented engineering methodology for the domain of Knowledge Management.

Unsoundness, *Non-Lucidity*, *Non-Laconicity* and *Incompleteness* violate what the philosopher of language H.P.Grice [12] 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.

(a) (b)

(c) (d)

Fig. 3. 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**.**

3 Language and Metamodel

The set of symbols that compose a language as well as the rules for forming valid combinations of these symbols constitute the language's syntax. In sentential languages, the syntax is first defined in terms of an alphabet (set of characters) that can be grouped into valid sequences forming words. This is called a lexical layer and it is typically defined using regular expressions. Words can be grouped into sentences according to precisely defined rules defined in a context-free grammar, resulting in an abstract syntax tree. Finally, these sentences are constrained by given context conditions.

In diagrammatic (graphical) languages, conversely, the syntax is not defined in terms of linear sequence of characters but in terms of pictorial signs. The set of available graphic modeling primitives forms the lexical layer and the language abstract syntax is typically defined in terms of a *metamodel*. Finally, the language metamodel is enriched by context-conditions given in some constraint description language, such as, OCL or first-order logic (FOL). In either case, context conditions are intended to constrain the language syntax by defining the set of correct (well-formed) sentences of the language. Some of these constraints are motivated by semantic considerations (laws of the domain being modeled as we shall see) while others are motivated by pragmatic issues [10]. Nevertheless, a metamodel is a description of the language's abstract syntax since it defines: (i) a set of constructs selected for the purpose of performing a specific (set of) task(s) and, (ii) a set of well-formedness rules for combining these constructs in order to create grammatically valid models in the language.

In the previous section, we advocate that the suitability of a language to create models in a given domain depends on how *"close"* the structure of the models constructed using that language resemble the structure of the domain abstractions they are supposed to represent. To put it more technically, a model $\mathcal{M}$ produced in a language $\mathcal{L}$ should be, at least, a homomorphism of the abstraction $\mathcal{A}$ that $\mathcal{M}$ represents. This evaluation can be systematically performed ultimately based on the analysis of the relation between the structure of a modeling language and the structure of a domain conceptualization.

What is referred by *structure of a language* can be accessed via the description of the specification of *conceptual model underlying the language*, i.e., a description of the worldview embedded in the language's modeling primitives. In [13], this is called the *ontological metamodel of the language*, or simply, the *ontology of the language*. From a philosophical standpoint, this view is strongly associated with Quine [14], who proposes that an ontology can be found in the *ontological commitment* of a given language, that is, the entities the primitives of a language commit to the existence of. For example, Peter Chen's Entity Relationship model [15] commits to a worldview that accounts for the existence of three types of things: *entity*, *relationship* and *attribute*.

This distinction of a metamodel as a pure description of a language's abstract syntax and as a description of the worldview underlying the language can be understood in analogy to the distinction between a design model and a conceptual model in information systems and software engineering. Whilst the latter is only concerned with modeling a view of the domain for a given application (or class of applications), the former is committed to translating the model of this view on the most suitable implementation according to the underlying implementation environment and also considering a number of non-functional requirements (e.g., security, fault-tolerance, adaptability, reusability, etc.). Likewise, the specification of the conceptual model underlying a language is the description of what the *primitives of a language* are able to represent in terms of real-world phenomena. In some sense (formally characterized in the next section), it is the representation of a conceptualization of the domain in terms of the language's vocabulary. In the *design* of a language, these conceptual primitives can be translated into a different set of primitives. For example, it can be the case that a conceptual primitive is not directly represented in the actual abstract syntax of a language, but its modeling capabilities (the real world concept underlying it) can be translated to several different elements in the language's abstract syntax due to non-functional requirements (e.g., pragmatics, efficiency). Nonetheless, the design of a language is responsible for guaranteeing that the language's syntax, formal semantics and pragmatics are conformant with this conceptual model. From now on, the *Modeling Language* icon depicted in figure 2 represents the specification of the conceptual model underlying the language, or what we shall name the *language metamodel specification***,** or simply the *language metamodel*.

The *structure of domain conceptualization* must also be made accessible through an explicit and formal description of the corresponding portion of reality in terms of a concrete artifact, which is termed here a *domain reference ontology*, or simply, a *domain ontology*. The idea is that a reference ontology should be constructed with the sole objective of making the best possible description of the domain in reality w.r.t. to a certain level of granularity and viewpoint. The notion of ontology as well as its role in the explicit representation of conceptualizations is discussed in depth and given a formal characterization in the next section.

4 Ontology, Metamodel and Conceptualization

Let us now return our attention to figure 2. A Modeling language can be seen as delimiting all possible specifications[1] which can be constructed using that language, i.e., all grammatically valid specifications of that language. Likewise, a conceptualization can be seen as delimiting all possible domain abstractions (representing state of affairs) which are admissible in that domain [16]. Therefore, for example, in a conceptualization of the domain of genealogy, there cannot be a domain abstraction in which a person is his own biological parent, because such a state of affairs cannot happen in reality. Accordingly, we can say that a modeling language which is truthful to this domain is one which has as valid (i.e., grammatically correct) specifications only those that represent state of affairs deemed admissible by a conceptualization of that domain. In the sequel, following [16], we present a formalization of this idea. This formalization compares conceptualizations as intentional structures and metamodels as represented by logical theories.

Let us first define a *conceptualization* C as follows:

Definition 1 (conceptualization): a conceptualization C is an intensional structure $\langle W,D,\mathfrak{R}\rangle$ such that W is a (non-empty) set of possible worlds, D is the domain of individuals and $\mathfrak{R}$ is the set of n-ary relations (concepts) that are considered in C. The elements $\rho \in \mathfrak{R}$ are intensional (or conceptual) relations with signatures such as $\rho^n:W \rightarrow \wp(D^n)$, so that each n-ary relation is a function from possible worlds to n-tuples of individuals in the domain. ■

For instance, we can have ρ accounting for the meaning of the natural kind apple. In this case, the meaning of apple is captured by the intentional function ρ, which refers to all instances of apples in every possible world.

Definition 2 (intended world structure): For every world $w \in W$, according to C we have an *intended world structure* S_wC as a structure $\langle D,R_wC\rangle$ such that $R_wC = \{\rho(w) \mid \rho \in \mathfrak{R}\}$. ■

More informally, we can say that every intended world structure S_wC is the characterization of some state of affairs in world w deemed admissible by conceptualization C. From a complementary perspective, C defines all the admissible state of affairs in that domain, which are represented by the set $S_c = \{S_wC \mid w \in W\}$.

Let us consider now a language $\mathcal{L}$ with a vocabulary V that contains terms to represent every concept in C.

Definition 3 (logical model): A logical model for $\mathcal{L}$ can be defined as a structure $\langle S,I\rangle$: S is the structure $\langle D,R\rangle$, where D is the domain of individuals and R is a set of extensional relations; $I:V \rightarrow D \cup R$ is an interpretation function assigning elements of D to constant symbols in V, and elements of R to predicate symbols of V. A model, such as this one, fixes a particular extensional interpretation of language $\mathcal{L}$. ■

Definition 3.4 (intensional interpretation): Analogously, we can define an intensional interpretation by means of the structure $\langle C,\mathfrak{I}\rangle$, where $C = \langle W,D,\mathfrak{R}\rangle$ is a conceptualization and $\mathfrak{I}:V \rightarrow D \cup \mathfrak{R}$ is an intensional interpretation function which assigns elements of D to constant symbols in V, and elements of $\mathfrak{R}$ to predicate symbols in V. ■

---

[1] We have so far used the term *model* instead of specification since it is the most common term in conceptual modeling. In this session, exclusively, we adopt the latter in order to avoid confusion with the term (logical) model as used in logics and tarskian semantics. A specification here is a syntactic notion; a logical model is a semantic one.

In [16], this intentional structure is named the *ontological commitment* of language $\mathcal{L}$ to a conceptualization C. We therefore consider this intensional relation as corresponding to the *represents* relation depicted in Ullmann's triangle in figure 1.

Definition 5 (ontological commitment): Given a logical language $\mathcal{L}$ with vocabulary V, an *ontological commitment* $K = \langle C, \Im \rangle$, a model $\langle S, I \rangle$ of $\mathcal{L}$ is said to be compatible with K if: (i) $S \in S_c$; (ii) for each constant c, $I(c) = \Im(c)$; (iii) there exists a world w such that for every predicate symbol p, I maps such a predicate to an admissible extension of $\Im(p)$, i.e. there is a conceptual relation ρ such that $\Im(p) = \rho$ and $\rho(w) = I(p)$. The set $I_k(\mathcal{L})$ of all models of $\mathcal{L}$ that are compatible with K is named the set of *intended models* of $\mathcal{L}$ according to K. ■

Definition 6 (logical rendering): Given a specification X in a specification language $\mathcal{L}$, we define as the logical rendering of X, the logical theory T that is the first-order logic description of that specification [17]. ■

In order to exemplify these ideas let us take the example of a very simple conceptualization C such that W = {w,w'}, D = {a,b,c} and $\Re$ = {person, father}. Moreover, we have that person(w) = {a,b,c}, father(w) = {a}, person(w') = {a,b,c} and father(w') = {a,b}. This conceptualization accepts two possible state of affairs, which are represented by the world structures $S_w C$ = {{a,b,c}, {{a,b,c},{a}} and $S_{w'} C$ = {{a,b,c}, {{a,b,c},{a,b}}. Now, let us take a language $\mathcal{L}$ whose vocabulary is comprised of the terms `Person` and `Father` with an underlying metamodel that poses no restrictions on the use of these primitives. In other words, the metamodel of $\mathcal{L}$ has the following logical rendering (T_1):

1. $\exists$x `Person(x)`
2. $\exists$x `Father(x)`

In this case, we can clearly produce a logical model of $\mathcal{L}$ (i.e., an interpretation that validates the logical rendering of $\mathcal{L}$) but that is not an intended world structure of C. For instance, the model D'={a,b,c}, person = {a,b}, father = {c}, and I(`Person`) = person and I(`Father`) = father. This means that we can produce a specification using $\mathcal{L}$ which model is not an *intended model* according to C.

Now, let us update the metamodel of language $\mathcal{L}$ by adding one specific axiom and, hence, producing the metamodel (T_2):

1. $\exists$x `Person(x)`
2. $\exists$x `Father(x)`
3. **$\forall$x `Father(x)` $\rightarrow$ `Person(x)`**

Contrary to $\mathcal{L}$, the resulting language $\mathcal{L}'$ with the amended metamodel T_2 has the desirable property that all *its valid specifications have logical models that are intended world structures of C.*

We can summarize the discussion so far as follows. A domain conceptualization C can be understood as describing the set of all possible state of affairs, which are considered admissible in a given universe of discourse U. Let V be a vocabulary whose terms directly correspond to the intensional relations in C. Now, let X be a *conceptual specification* (i.e., a concrete representation) of universe of discourse U in terms of the vocabulary V and let T_X be a logical rendering of X, such that its axiomatization constrains the possible interpretations of the members of V. We call X (and T_X) an *ideal ontology* of U according to C iff the logical models of T_X describe all and only state of affairs which are admitted by C.

The relationships between language vocabulary, conceptualization, ontological commitment and ontology are depicted in figure 4 below. This use of the term ontology is strongly related to the third sense (D3) in which the term is used in philosophy, i.e. as "*a theory concerning the kinds of entities and specifically the kinds of abstract entities that are to be admitted to a language system*" (section 1).

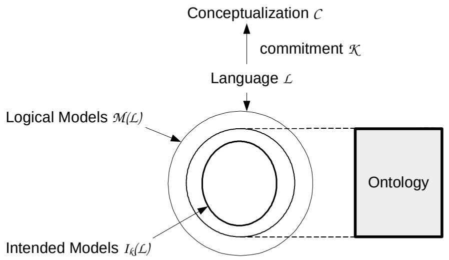

Fig. 4. Relations between language (vocabulary), conceptualization, ontological commitment and ontology

The logical theory (T_2) described above is, thus, an example of an ontology for the person/father toy conceptualization. As pointed out in [16], ontologies cannot always be ideal and, hence, a general definition for an (non-ideal) ontology must be given: *An ontology is a conceptual specification that describes knowledge about a domain in a manner that is independent of epistemic states and state of affairs. Moreover, it intends to constrain the possible interpretations of a language's vocabulary so that its logical models approximate as well as possible the set of intended world structures of a conceptualization C of that domain.*

According to this criterion of accuracy, we can therefore give a precise account for the quality of a given ontology. Given an ontology O_L and an ideal ontology O_C, the quality of O_L can be measured as the distance between the set of logical models of O_L and O_C. In the best case, the two ontologies share the same set of logical models. In particular, if O_L is the specification of the ontological metamodel of modeling language $\mathcal{L}$, we can state that if O_L and O_C are *isomorphic* then they also share the same set of possible models. It is important to emphasize the relation between the possible models of O_L and the completeness of language $\mathcal{L}$ (in the technical sense briefly discussed in section 2). There are two ways in which incompleteness can impact the quality of O_L: firstly, if O_L (and thus $\mathcal{L}$) does not contain concepts to fully characterize a state of affairs, it is possible that the logical models of O_L describe situations that are present in several world structures of C. In this case, O_L is said to *weakly* characterize C [16], since it cannot guarantee the reconstruction of the relation between worlds and extensional relations established by C; secondly, if the representation of a concept in O_L is underspecified, it will not contain the axiomatization necessary to exclude unintended logical models. As an example of the latter, we can mention the incompleteness of UML class diagrams w.r.t. classifiers and part-whole relations discussed in [10]. In summary, we can state that an ideal ontology O_C for a conceptualization C of universe of discourse U can be seen as the specification of the ontological metamodel for an ideal language to represent U according to C. For this reason, the adequacy of a language $\mathcal{L}$ to represent phenomena in U can be systematically evaluated by comparing $\mathcal{L}$'s metamodel specification with O_C. This idea is illustrated in figure 5 below.

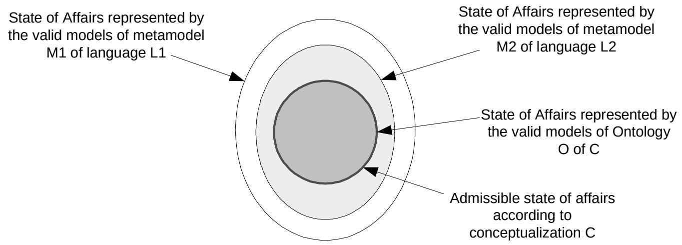

Fig. 5. Measuring the degree of *domain appropriateness* of modeling languages via an ontology of a conceptualization of that domain.

By including formula (3), (T_1) is transformed into an ideal ontology (T_2) of C. One question that comes to the mind is: *How can one know that?* In other words, how can we systematically design an ontology O that is a better characterization of C. There are two important points that should be called to attention. The first point concerns the language that is used in the representation of these specifications, namely, that of standard predicate calculus. The formula added to (T_1) to create (T_2) represents a subsumption relation between Person and Father. Subsumption is a basic primitive in the group of the so-called *epistemological* *languages*, which includes languages such as UML, EER and OWL. It is, in contrast, absent in ontological neutral logical languages such as predicate calculus. By using a language such as OWL to represent a conceptualization of this domain, a specification such as the one in figure 6 should be produced (which reads “Father *is-a* Person”, or in other words, the Father concept is subsumed by the Person concept). In this model, the third axiom would be automatically included through the semantics of the metamodeling language. Therefore, if a suitable ontology modeling language is chosen, its primitives incorporate an axiomatization, such that the specifications (ontologies) produced using this language will better approximate the intended models of a conceptualization C.

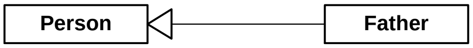

Fig. 6. Example of a subsumption relation in UML

The second point that should be emphasized is related to the question: *how are the world structures that are admissible to C determined?* The rationale that we use to decide that are far from arbitrary, but motivated by the laws that govern the domain in reality. In [18], the philosopher of science Mario Bunge defines the concepts of a *state space* of a thing[2], and a subset of it, which he names a *nomological state space* of a thing. The idea is that among all the (theoretically) possible states a thing can assume, only a subset of it is lawful and, thus, is actually possible. Additionally, he defends that the only really possible *facts* involving a thing are those that abide by laws, i.e., those delimited by the nomological state space of thing. As a generalization, if an actual state of affairs consists of facts [19], then the set of possible state of affairs is determined by a *domain nomological state space*. In sum, possibility is not by any means defined arbitrarily, but should be constrained by the set of laws that constitute reality. For example, it is law of the domain (in reality) that every Father is a Person. The specification (T_2) is an ideal ontology for C because it includes the representation of this law of this domain via the subsumption relation between the corresponding representations of father and person. Conversely, if C included a world structure in which this law would be broken, the conceptualization itself would not be truthful to reality. To refer once more to Ullmann’s triangle (figure 1), the relation between a conceptualization C and the *domain nomological state space* is that relation of *abstracts* between conceptualization and reality.

Now, to raise the level of abstraction, we can also consider the existence of a meta-conceptualization C_0, which defines the set of all domain conceptualizations such as C that are truthful to reality. Our main objective is to define a general ontology representation language $\mathcal{L}_0$ that can be used to produce domain ontologies such as O_C, i.e., a language whose primitives include theories that help in the formal characterization of a domain-specific language $\mathcal{L}$, restricting its logical models to those deemed admissible by C. In order to do this, we have to include primitives in language $\mathcal{L}_0$ that represent the laws that are used to define the nomological world space of meta-conceptualization C_0. In this case, these are the general laws that describe reality, and describing these laws is the very business of *formal ontology* in philosophy.

In summary, we defend that the ontology underlying a general ontology representation language $\mathcal{L}_0$ should be a meta-ontology that describes a set of real-world categories that can be used to talk about reality, i.e., ontology in the sense (D2) of section 1. Likewise, the axiomatization of this meta-ontology must represent the laws that define that nomological world space of reality. This meta-ontology, when constructed using the theories developed by *formal ontology* in philosophy, is named a *foundational ontology*.

We can summarize the main points of this latter discussion as follows. A domain such as genealogy is what is named in the literature a *material domain* [20] and a language designed to model phenomena in this domain is called a *domain-specific language*. According to the language evaluation framework mentioned in section 2, we can provide the following definition for an ideal language to represent phenomena in a given domain:

A language is ideal to represent phenomena in a given domain if the metamodel of this language is isomorphic to the ideal ontology of that domain, and the language only has as valid specifications those whose logical models are exactly the logical models of the ideal ontology.

---

[2] The word Thing is used by Bunge in a technical sense, which is synonymous to the notion of *substantial individual* as used in [10].

This principle should hold not only for domain-specific languages, whose metamodels should be isomorphic to some ontology of a material domain, but also for domain-independent languages and, in particular, for general ontology representation languages that can be used to create domain ontologies in different material domains. To be consistent with the position defended here, a language $\mathcal{L}_0$ used to represent individual domain ontologies should also be based on a conceptualization, but in this case, a meta-conceptualization, which is represented by a *Foundational Ontology*. This idea is illustrated in figure 7 below.

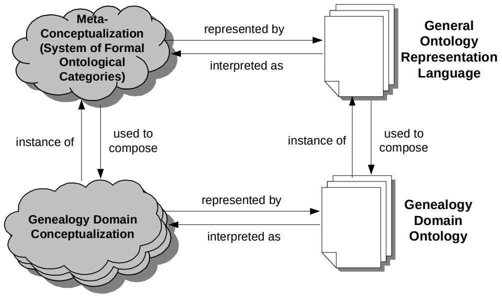

Fig. 7. Relations between Material Domain Conceptualization, Domain Ontologies, General Ontology Representations Languages and their Meta-conceptualization as a formal system of ontological categories.

5 Towards a suitable General Ontology Representation Language: *The Ontological Level* Revisited

When a general ontology representation language is constrained in such a way that its intended models are made explicit, it can be classified as belonging to the *ontological level*. This notion has been proposed by Nicola Guarino in [21], in which he revisits Brachman's classification of knowledge representation formalisms [22].

In Brachman's original proposal, the modeling primitives offered by knowledge representation formalisms are classified in four different levels, namely: *implementation*, *logical*, *conceptual* and *linguistic* levels.

In the logical level, we are concerned with the predicates necessary to represent the concepts of a domain and with evaluating the truth of these predicates for certain individuals. The basic primitives are propositions, predicates, functions and logical operators, which are extremely general and ontologically neutral. For instance, suppose we want to state that a red apple exists. In predicate calculus we would write down a logical formula such as (F_1) $\exists \mathtt{x}(\mathtt{apple(x)} \wedge \mathtt{red(x)})$. Although this formula has a precise semantics, the real-world interpretation of a predicate occurring in it is completely arbitrary, since one could use it to represent a property of a thing, the kind the thing belongs to, a role played by the thing, among other possibilities. In this example, the predicates apple and red are put in the same logical footing, regardless of the nature of the concept they represent and the importance of this concept for the qualification of predicated individual. Logical level languages are neutral w.r.t. ontological commitments and it is exactly this neutrality that makes logic interesting to be used in the development of scientific theories. However, it should be used with care and not directly in the development of ontologies, since one can write perfectly correct logical formulas, but which are devoid of ontological interpretation. For example, the entailment relation has no ontic correlation. Moreover, while one can negate a predicate or construct a formula by a disjunction of two predicates, in reality, there are neither negative nor alternative entities [18].

In order to improve the "flatness" of logical languages, Brachman proposes the introduction of an *epistemological level* on top of it, i.e., between the logical and conceptual levels in the original classification. Epistemology is the branch of philosophy that studies *"the nature and sources of knowledge"*. The interpretation taken by Brachman and many other of the logicist tradition in AI is that knowledge consists of propositions, whose formal structure is the source of new knowledge. Examples of representation languages belonging to this level include Brachman's own KL-ONE [23] and its derivatives (including the semantic web languages OIL, DAML, DAML+OIL, RDFS, OWL) as well as object-based and frame-based modeling languages such as EER, LINGO [24] and UML. The rationale behind the design of epistemological languages is the following: (i) the languages should be designed to capture interrelations between pieces of knowledge that cannot be smoothly captured in logical languages; (ii) they should offer *structuring* mechanisms that facilitate understanding and maintenance, they should also allow for economy in representation, and have a greater computational efficiency than their logical counterparts; (iii) finally, modeling primitives in these languages should represent structural connections in our knowledge needed to justify conceptual inferences in a way that is independent of the meaning of the concepts themselves.

Indeed languages such as UML, ER, LINGO and OWL offer powerful structuring mechanisms such as classes, relationships (attributes) and subclassing relations. However, if we want to impose a certain *structure* in the representation of formula (F_1), in a language such as UML, we would have to face the following structuring choices: (a) consider that there are instances of apples that can posses the property of being red or, (b) consider that there are instances of red things that can have the property of being apples. Formally we can state either that (F_2) `∃x:Apple.red(x)` as well as (F_3) `∃x:Red.apple(x)`, and both these many-sorted logic formalizations are equivalent to the previous one-sorted axiom. However, each one contains an implicit structuring choice for the sort of the things we are talking about.

The design of epistemological languages puts a strong emphasis on the inferential process, and the study of knowledge is limited to its form, i.e., it is *"independent of the meaning of the concepts themselves"*. Therefore, the focus of these languages is more on formal reasoning than on (formal) representation. Returning to our example, although the representation choice (b) seems to be intuitively odd, there is nothing in the semantics of a UML class or an OWL concept that prohibits any unary predicate such as red or tall to be modeled as such. In other words, since in epistemological languages the semantics of the primitive “sort” is the same as its corresponding unary predicate, the choice of which predicates correspond to sorts is completely left to the user.

In [21], Guarino points out that structuring decisions, such as this one, should not result from heuristic considerations but instead should be motivated and explained in the basis of suitable *ontological distinctions*. For instance, in this case, the choice of Apple as the sort (a) can be justified by the meta-properties that we are ascribed to the term by the *intended meaning* that we give to it. The ontological difference between the two predicates is that Apple corresponds to a Natural *Kind* whereas Red corresponds to an Attribution or a *Mixin* [10]. Whilst the former applies necessarily to its instances (an apple cannot cease to be an apple without ceasing to exist), the latter only applies contingently. Moreover, whilst the former supplies a *principle of identity*[3] for its instances, i.e., a principle through which we judge if two apples are numerically the same, the latter cannot supply one. However, it is not the case that an object could subexist without obeying a principle of identity [25], an idea which is defended both in philosophical ontology (e.g., Quine's dicto *"no entity without identity"* [14]), and in conceptual modeling (e.g., Chen's design rational for ER [15]). Consequently, the structuring choice expressed in (F_3) cannot be justified.

In addition to supporting the justified choice for structuring decisions, the ontological level has important practical implications from a computational point of view. For instance, one can exploit the knowledge of which predicates hold necessarily (and which are susceptible to change) in the design and implementation of more efficient update mechanisms. Finally, there are senses in which the term Red can be said to hold necessarily (e.g., *"scarlet is a type of red"* referring to a particular shade of color), and senses in which it carries a principle of identity to its instances (e.g., *"John is a red"* - meaning that *"John is a communist"*). The choice of representing Red as a *Mixin* in the aforementioned representation makes explicit the intended meaning of this predicate, ruling out these two other possible interpretations. In epistemological and logical languages, conversely, the intended meaning of a predicate relies on our understanding of the natural language label used. As this example makes explicit, an ontologically well-founded modeling language should commit to a system of ontological meta-level categories that include, for instance, *Kinds*, *Roles*, and *Mixin* as distinctions which further qualify and make precise and explicit the real-world semantics of the terms used in domain representations. An example a foundational ontology that comprises such a system of categories is discussed in the next section.

---

[3] For an extensive discussion on *kinds, attributions* and *principles of identity* as well as their importance for the practice of conceptual modeling we refer to [10].

6 Domain-Independent and Domain-Specific Languages: An Illustrative Example

The language evaluation and design framework briefly discussed in section 2 can be applied both at the level of material domains (e.g., genomics, archeology, multimedia, fishery, law, etc.) and corresponding domain-specific modeling languages, and at the (meta) level of a domain-independent (meta) conceptualization that underpins a general conceptual (ontology) modeling language. In [10], we have developed a Foundational Ontology named **UFO (Unified Foundational Ontology)** which can be used as theoretically sound basis for evaluating and redesigning conceptual modeling languages, in general, and ontology representation languages in particular. In figure 8, we illustrate a small excerpt of this foundational ontology that contains a typology of universals (roughly classes, types).

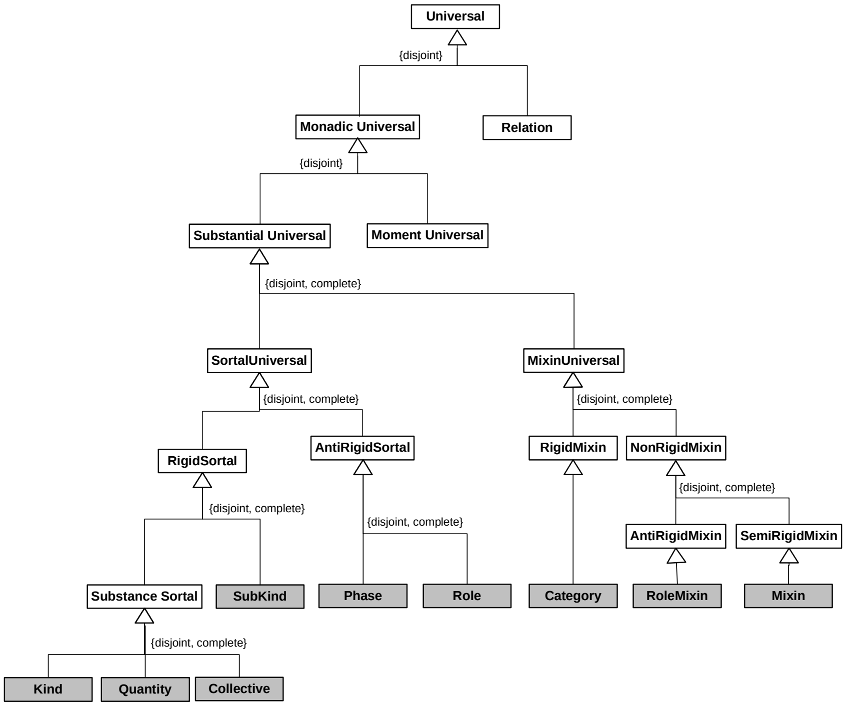

Fig 8. Excerpt of the Foundational Ontology UFO depicting a typology of universals [10].

In [10], we used this fragment of the UFO ontology to evaluate and re-design the portion of UML dealing with classifiers for the purpose of conceptual modeling and ontology representation. The re-designed UML 2.0 metamodel resulting from this process is depicted in figure 9. This metamodel describes the abstract syntax of (part of) a general ontology representation language. The UML profile implementing this metamodel is illustrated in figure 10, in which it is used to represent an ontology for the genealogy material domain. In [10], we also used this domain ontology and the framework discussed in section 2 to systematically design a domain-specific modeling language in the domain of Genealogy (named hereafter L_1). The modeling primitives of L_1 are depicted in figures 11. Figure 12 presents examples of invalid (fig.12.a-c) and valid models (fig.12.d) produced using that language.

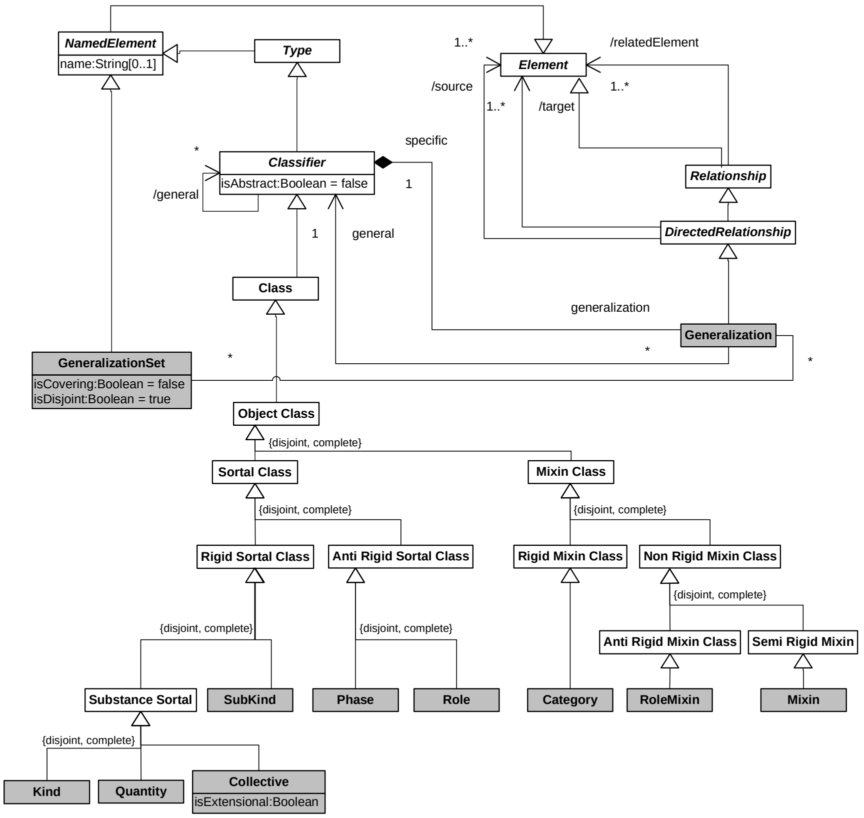

Fig 9. Redesigned UML 2.0 metamodel according to the Foundational Ontology of Fig.8.

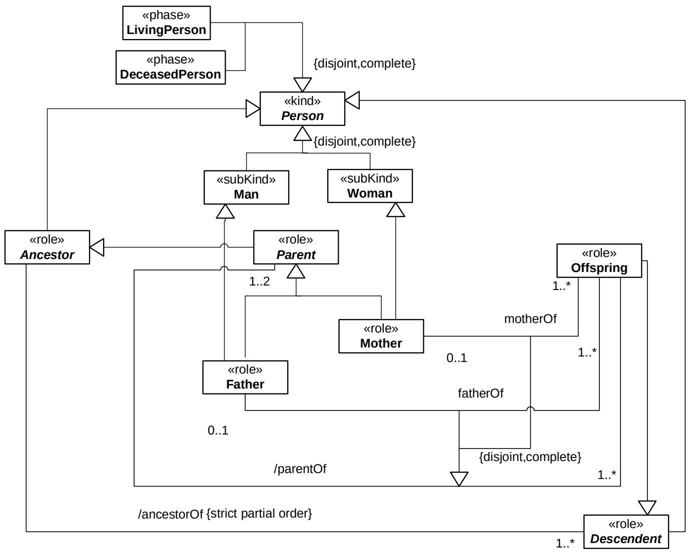

Fig. 10. An ontology for the genealogy domain.

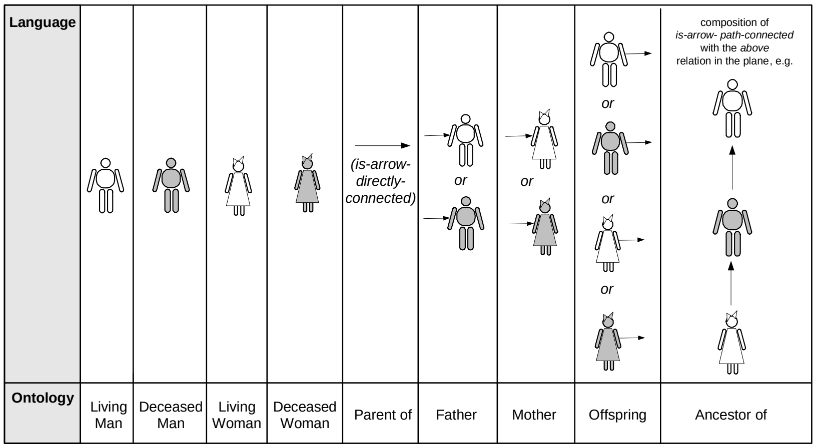

Fig 11. Domain Concepts and their representing modeling primitives in language L_1.

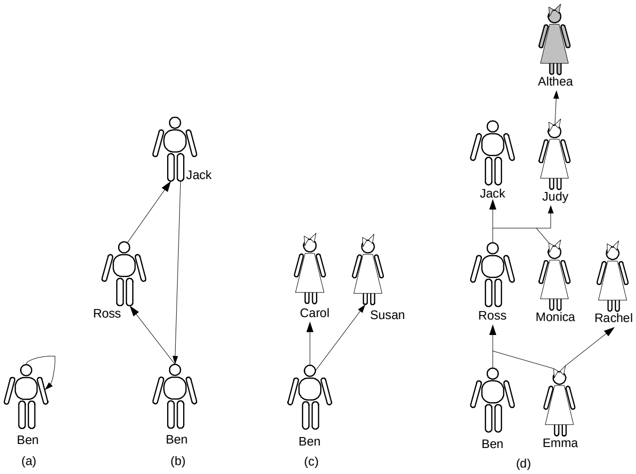

Fig 12. (a-c) Examples of invalid models and (d) of a valid model in L_1.

By instantiating the pattern of figure 2 to the whole example discussed so far we obtain the correspondences depicted in figures 13 and 14. The ontology O of figure 10 is a concrete representation of a given conceptualization of the genealogy domain. In this case, we have the ideal situation that the metamodel of language L_1 is isomorphic to this ontology. The genealogy concepts represented in O are used to articulate models of individual state of affairs in reality. A specification in language L_1 (such as the one of figure 12.d) is a concrete artifact representing one of these models. Since L_1 is an ideal language to represent the genealogy domain according to the ontology of figure 10, the only grammatically valid models of this language are the ones which represent abstractions which are deemed acceptable according to that ontology. For this reason, models such as the ones that represent abstractions in which someone is his own father/ancestor (12.a), or a father/ancestor of one of his ancestors (12.b) are invalid models in this language. Finally, a domain ontology such as the one just discussed is also a concrete artefact (a model), and as much as the models in figure 12, it must be represented in some modeling language. An example of such a language is the version of UML (the UML profile) used in figure 10. This modelling language has a metamodel (figure 9) which is isomorphic to the foundational ontology whose fragment is depicted in figure 8. Here once more, the grammatically valid models (domain ontologies) according to this UML profile are only those that represent (domain) conceptualizations which are deemed accepted by the UFO ontology. So, for instance, one cannot produce in this language a conceptualization in which (i) *a Role is supertype of a Kind*, or that (ii) *an Object Class is not a subkind of exactly one Kind*[4].

---

[4] These two constraints (i) and (ii) have been formally proved in [10].

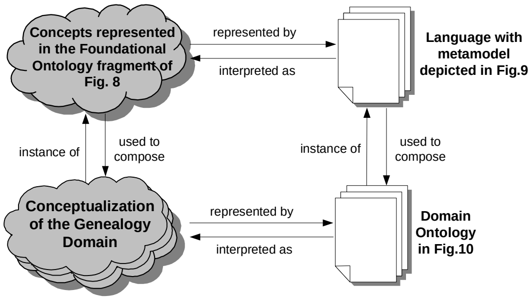

Fig 13. Instantiating the pattern of figure 2 for a domain-independent meta-conceptualization.

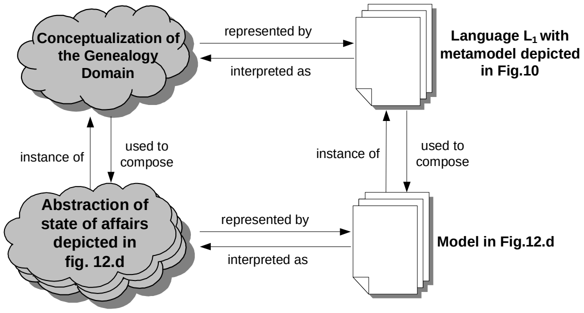

Fig 14. Instantiating the pattern of figure 2 for the domain of genealogy.

7 Reference and Lightweight Ontologies

Conceptual Models vary in the way they manage to represent an associated conceptualization. An ontology such as the one in figure 10 is more accurate than if it were represented in ER, OWL, LINGO or standard UML. This is because the modeling profile used in that model commits to a much richer meta-ontology than the ones underlying these other languages. As a consequence, to formally characterize its ontological distinctions, a formal language with higher expressiveness is needed. When the stereotyped modeling primitives of this profile are used, an axiomatization in the language of *intensional modal logics* is incorporated in the resulting specification, constraining the interpretation of its terms[5]. Quantified Intensional modal logics are more expressive than, for example, a $SHOIN(D_n)$ descriptions logics, which is the language behind the formalization of OWL. In contrast, a language such as OWL has been carefully designed to maintain interesting properties such as computational tractability and decidability, which are properties that are in general absent in more expressive languages. Likewise, LINGO was designed to facilitate the translation to Object-Oriented implementations. Properties such as computational efficiency and easiness of translation to implementation platforms have been recognized as important properties to application areas of ontology in computer science such as the Semantic Web initiative and Domain Engineering [24]. Therefore, in Ontology Engineering, the following tradeoff must be recognized. On one side we need a language that commits to a rich foundational ontology. This meta-ontology, however, will require the use of highly expressive formal languages for its characterization, which in general, are not interesting from a computational point of view. On the other side, languages that are efficient computationally, in general, do commit to a suitable meta-conceptualization. The obvious question is then: how can we design a suitable general ontology representation language according to these conflicting requirements?

The position advocated here is analogous to the one defended in [26], namely, that we actually need two classes of languages. We explain this position by once more making use of an analogy to the engineering processes in the disciplines of Software and Information Systems Engineering. In both

---

[5] The complete formal semantics of this profile in a system of Modal Logics with Sortal Quantification is presented in [10].

disciplines, there is a clear distinction between Conceptual Modeling, Design and Implementation. In Conceptual Modeling, a solution-independent specification is produced whose aim is to make a clear and precise description of the domain elements for the purposes of communication, learning and problem-solving. In the Design phase, this conceptual specification is transformed in a design specification by taking into consideration a number of issues ranging from architectural styles, non-functional quality criteria to be maximized, target implementation environment, etc. The same conceptual specification can potentially be used to produce a number of (even radically) different designs. Finally, in the Implementation phase, a design is coded in a target language to be then deployed in a computational environment. Again, from the same design, a number of different implementations can be produced. Design, thus, bridges Conceptual Modeling and Implementation.

We here defend an analogous principle for Ontology Engineering. On one hand, in a conceptual modeling phase in Ontology Engineering, highly-expressive languages should be used to create strongly axiomatized ontologies that approximate as well as possible to the ideal ontology of the domain. The focus on these languages is on representation adequacy, since the resulting specifications are intended to be used by humans in tasks such as communication, domain analysis and problem-solving. The resulting domain ontologies, named *reference ontologies* in [16], should be used in an *off-line* manner to assist humans in tasks such as meaning negotiation and consensus establishment. On the other hand, once users have already agreed on a common conceptualization, versions of a reference ontology can be created. These versions have been named in the literature *lightweight ontologies.* Contrary to reference ontologies, lightweight ontologies are not focused on representation adequacy but are designed with the focus on guaranteeing desirable computational properties. Examples of languages suitable for lightweight ontologies include OWL and LINGO. An example of an ontology representation language that is suitable for reference ontologies is the UML profile briefly illustrated in the previous section (as demonstrated in [10]). It is important, nonetheless, to highlight that, as discussed in the previous section, languages such as OWL and LINGO are epistemological level languages and, thus, naming them ontology representation languages is actually a misnomer. Finally, a phase is necessary to bridge the gap between the conceptual modeling of references ontologies and the coding of these ontologies in terms of specific lightweight ontology languages. Issues that should be addressed in such a phase are, for instance, determining how to deal with the difference in expressivity of the languages that should be used in each of these phases, or how to produced lightweight specifications that maximize specific non-functional requirements (e.g., evolvability vs. reasoning performance).

Finally, the importance of reference ontologies has been acknowledged in many cases in practice. For instance, [27] illustrates examples of semantic interoperability problems that can pass undetected when interoperating lightweight ontologies. Likewise, [28] discusses how a principled foundational ontology can be used to spot inconsistencies and provide solutions for problems in lightweight biomedical ontologies. As a final example, the need for methodological support in establishing precise meaning agreements is recognized in the *Harvard Business Review* report of October 2001, which claims that *"one of the main reasons that so many online market makers have foundered [is that] the transactions they had viewed as simple and routine actually involved many subtle distinctions in terminology and meaning"*.

8 Summary

In the sequel, we summarize the most important points defended in this article:

1. *Formal Ontology*, as conceived by Husserl, is part of the discipline of Ontology in philosophy (sense D1), which is, in turn, the most important branch of metaphysics;
2. Formal Ontology aims at developing general theories that accounts for aspects of reality that are not specific to any field of science, be it physics or conceptual modeling (sense D2);
3. These theories describe knowledge about reality in a way, which is independent of language, of particular states of affairs (states of the world), and of epistemic states of knowledgeable agents. In this article, these language independent theories are named (meta) *conceptualizations*. The representation of these theories in a concrete artifact is a *foundational ontology*;

4. A Foundational Ontology is domain-independent Reference Ontology. Reference Ontologies try to characterize as accurately as possible the conceptualization they commits to.
5. Foundational ontologies in the philosophical sense can be used to provide real-word semantics for general *ontology representation languages* and to constrain the possible interpretations of their modeling primitives. Conversely, a suitable ontology representation language should commit to a system of ontological distinctions (in the philosophical sense), i.e., they should truly belong to the *Ontological Level*.
6. An ontology can be seem as the *metamodel* specification for an ideal language to represent phenomena in a given domain in reality, i.e., a language which only admits specifications representing possible state of affairs in reality (related to sense D3).
7. Suitable general conceptual modeling languages can be used in the development of reference *domain ontologies*, which, in turn, among many other purposes, can be used to characterize the vocabulary of *domain-specific languages*.
8. The discipline of Ontology Engineering should account for two classes of languages with different purposes: (i) on one hand, it needs well-founded ontology representation languages focused on representation adequacy regardless of the consequent computational costs, which is not actually a problem since the resulting model is targeted at human users; (ii) On the other hand, it needs lightweight representation languages with adequate computational properties to guarantee their use as codification alternatives for the reference ontologies produced in (i).
9. The name ontology representation languages when applied to the so-called Semantic Web languages is a misnomer, since these languages are motivated by epistemological and computational concerns, not ontological ones.

Acknowledgements. The author would like to thank Gerd Wagner, Nicola Guarino, Luís Ferreira Pires, Marten van Sinderen, Renata S.S. Guizzardi and Chris Vissers for fruitful discussions and for providing valuable input to the issues of this article.

References

1. Merriam-Webster Dictionary, [online: www.m-w.com], captured in 2004.
2. Mealy, G. H. ‘Another Look at Data’, Proc. of the Fall Joint Computer Conference, Anaheim, California (AFIPS Conference Proceedings, Volume 31), Washington, DC: Thompson Books, London: Academic Press, 525–534, 1967.
3. Quine, W. V. O. ‘On What There Is’, as reprinted in From a Logical Point of View, New York: Harper & Row, 1953.
4. Hayes P. ‘The Naïve Physics Manifesto’, In D. Ritchie (Ed.) Expert Systems in Microeletronics age. Edinburgh University Press, pp 242-270, 1978.
5. Hayes, P. ‘Naïve Physics I: Ontology for Liguids’, in Hobbs & Moore, pp.71-107, 1985.
6. Ullmann, S. ‘Semantics: An Introduction to the Science of Meaning’, Basil Blackwell, Oxford, 1972.
7. Ogden, C. K., Richards, I. A. ‘The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism’, New York: Harcourt Brace Jovanovich, 1923.
8. de Saussure, F. ‘Course in General Linguistics’, Roy Harris (trans.), Open Court Publishing Company, 1986 (original from 1916).
9. Baldinger, K. ‘Semantic Theory: Towards a Modern Semantics’, Palgrave Macmillan, 1980.
10. Guizzardi, G. “Ontological Foundations for Structural Conceptual Models”, Telematica Instituut Fundamental Research Series no. 15, Universal Press, The Netherlands, 2005, ISBN 90-75176-81-3.
11. 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, 2007 (forthcoming)..
12. 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.
13. Milton, S.K., Kamierczak, E. ‘An Ontology of Data Modeling Languages: A Study Using a Common-Sense Realistic Ontology’, Journal of Database Management, Special Issue on Ontological Analysis, Evaluation, and Engineering Of Business Systems Analysis Methods, 18 pages, 2004.

14. Quine, W. V. O. ‘Ontological relativity’, and Other Essays, New York: Columbia University Press, 1969.
15. Chen. P. ‘The entity-relationship model: Towards a unified view of data’, ACM Transactions on Database Systems 1(1), 1976.
16. Guarino, N. ‘Formal Ontology and Information Systems’, In Guarino, N. (ed.) Formal Ontology in Information Systems, Proceedings of the International Conference on Formal Ontology and Information Systems (FOIS), Trento, Italy, June 6-8. IOS Press, Amsterdam: pp. 3-15, 1998.
17. Ciocoiu, M., Nau D. ‘Ontology-Based Semantics’, In: Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR'2000), Breckenbridge, Colorado, April 12-17, 2000.
18. Bunge M. ‘Ontology I: The Furniture of the World’, Treatise on Basic Philosophy, vol. 3, D. Reidel Publishing, New York, 1977.
19. Armstrong, D.M. ‘A World of State of Affairs’, Cambridge Studies in Philosophy, Cambridge University Press, 1997.
20. Smith, B. ‘Logic and Formal Ontology’, in J. N. Mohanty and W. McKenna, eds., Husserl's Phenomenology: A Textbook, Lanham: University Press of America, 29-67, 1989.
21. Guarino, N. ‘The Ontological Level’, In R. Casati, B. Smith and G. White (eds.), Philosophy and the Cognitive Science. Holder-Pivhler-Tempsky, Vienna: pp. 443-456, 1994.
22. Brachman, R. J. ‘On the Epistemological Status of Semantic Networks’, In N. V. Findler (Ed.), Associative Networks: Representation and Use of Knowledge by Computers.Academic Press, 1979.
23. Brachman, R. J. and J. G. Schmolze ‘An Overview of the KL-ONE Knowledge Representation System’, Cognitive Science 9, 171-216. Carrara, M. 1992. Identità e persona nella riflession, 1985.
24. Falbo, R. A., Guizzardi, G., Duarte, K. C. ‘An Ontological Approach to Domain Engineering’, In Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering (SEKE), Ischia, Italy, 2002.
25. van Leeuwen, J. ‘Individuals and sortal concepts : an essay in logical descriptive metaphysics’, PhD Thesis, University of Amsterdam, 1991.
26. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A. ‘Ontology Library’, WonderWeb Deliverable D18, 2003.
27. Guizzardi, G. “The Role of Foundational Ontology for Conceptual Modeling and Domain Ontology Representation”, Keynote Paper, 7th International Baltic Conference on Databases and Information Systems, Vilnius, Lithuania, 2006.
28. Fielding, J.M.; Simon, J.; Ceusters, W.; Smith, B. ‘Ontological Theory for Ontology Engineering’. in: Proceedings of KR 2004, Ninth International Conference on the Principles of Knowledge Representation and Reasoning, Whistler, Canada, 2004.