Towards Ontological Foundations for Conceptual Modeling:
The Unified Foundational Ontology (UFO) Story

Giancarlo Guizzardi\textsuperscript{1,2}, Gerd Wagner\textsuperscript{3}, João Paulo Andrade Almeida\textsuperscript{1}, Renata S.S. Guizzardi\textsuperscript{1}

\textsuperscript{1}Ontology and Conceptual Modeling Research Group (NEMO), Federal University of Espírito Santo, Vitória, Brazil \{gguizzardi, jpalmeida, rguizzardi\}@inf.ufes.br

\textsuperscript{2}Laboratory for Applied Ontology (LOA), Institute for Cognitive Science and Technology (ISTC) Italian National Research Council (CNR), Trento, Italy

\textsuperscript{3}Institute of Informatics, Brandenburg University of Technology, Germany G.Wagner@tu-cottbus.de

\textbf{Abstract.} This paper describes a long-term research program on developing ontological foundations for conceptual modeling. This program, organized around the theoretical background of the foundational ontology UFO (Unified Foundational Ontology), aims at developing theories, methodologies and engineering tools with the goal of advancing conceptual modeling as a theoretically sound discipline but also one that has concrete and measurable practical implications. The paper describes the historical context in which UFO was conceived, briefly discusses its stratified organization, and reports on a number of applications of this foundational ontology over more than a decade. In particular, it discusses the most successful application of UFO, namely, the development of the conceptual modeling language OntoUML. The paper also discusses a number of methodological and computational tools, which have been developed over the years to support the OntoUML community. Examples of these methodological tools include ontological patterns and anti-patterns; examples of these computational tools include automated support for pattern-based model construction, formal model verification, formal model validation via visual simulation, model verbalization, code generation and anti-pattern detection and rectification. In addition, the paper reports on a variety of applications in which the language as well as its associated tools have been employed to engineer models in several institutional contexts and domains. Finally, it reflects on some of these lessons learned by observing how OntoUML has been actually used in practice by its community and on how these have influenced both the evolution of the language as well as the advancement of some of the core ontological notions in UFO.

\textbf{1. Introduction}

This paper describes a long-term research program on developing ontological foundations for conceptual modeling. This research program aims at developing theories, methodologies and engineering tools (including modeling languages, patterns, anti-patterns as well as computational tools) with the goal of advancing conceptual modeling as a theoretically sound discipline with concrete and measurable practical implications. For over a decade, our research program has been organized around the theoretical background of a foundational ontology termed UFO (Unified Foundational Ontology) and, frequently, materialized through the development of number of results associated with an ontologically well-founded conceptual modeling version of UML termed OntoUML.

In section 2, we start by briefly describing the historical context in which UFO was conceived. This has the purpose of highlighting the goals of the project and contextualizing the rationale behind fundamental ontological choices made in the design of UFO. In section 3, we briefly discuss the layers (or strata) comprising UFO and the ontological micro-theories comprising each of them. This section also reports on a number of applications of this foundational ontology over more than a decade, chiefly, related to the ontological analysis, (re) design and integration of different modeling languages, reference models and standards. In section 4, we discuss the most successful application of UFO, namely, the development of the OntoUML language. The section briefly discusses the design of this language and reports on a number of applications in which the language as well as its associated methodological and computational tools have been employed to engineer models in a variety of institutional contexts and domains. In addition, the section elaborates on the development of a model-based computational environment, which has been developed to support the OntoUML community in terms of pattern-based model construction, code generation, formal verification, formal validation via visual simulation, model verbalization and anti-pattern detection and rectification. Given the diffusion of the language, over the years, we have managed to assemble a model repository containing
OntoUML models in different domains. The observation of how the language was used in practice in the construction of these models proved to be a fruitful source of empirical knowledge about the actual notions that needed to be accounted for in an ontology-driven structural conceptual modeling language. Section 5 concludes the paper by reflecting on some of these lessons learned and how they have influenced both the evolution of the language of some core ontological notions in UFO.

2. A Brief Historical Background

In 2002, the first two authors have started a research program of conducting ontological analysis of conceptual modeling languages and standards with the goal of developing sound ontological foundations for conceptual modeling languages (Guizzardi et al., 2002a, 2002b). Our program was motivated by the strong belief that any serious scientific discipline should have well-established and explicitly defined metaphysical foundations (Bunge, 1977) or that, as nicely put by (Collier, 1994): “the opposite of ontology is not non-ontology, but bad ontology”. In other words, any representation system that has some real-world semantics (as opposed to mere formal semantics) makes an ontological commitment (Guizzardi & Halpin, 2008). This research program was motivated by the belief that controlling and defending a particular ontological commitment is essential for the progress of a scientific discipline.

In the realm of conceptual modeling, the idea that “data are fragments of a theory of the real-world and data processing is about manipulating models of such a theory” was there since Mealy’s seminal paper entitled ‘Another Look on Data’ (Mealy, 1967). In fact, Mealy’s paper seems to contain the first mention of the term ‘ontology’ in the computer and information science literature. A number of fundamental conceptual modeling issues of an ontological nature were also discussed in Bill Kent’s book ‘Data and Reality’ (Kent, 1978). This book brought attention to issues like identity, unity and classification, and started exposing the subtleties of fundamental conceptual modeling constructs such as relationships.

However, neither Mealy’s paper nor Kent’s book tried to actually develop comprehensive ontological foundations for conceptual modeling. Perhaps the first corpus of work to attempt that goal was reported in the series of publications initiated by Yair Wand, Ron Weber and colleagues (Wand & Weber, 1989, 1990; Weber, 1997) in the late 80’s. Instead of developing a new ontology themselves, Wand and Weber proposed an adaptation of the ontological theory put forth by the Argentinean physicist and philosopher of science Mario Bunge. The result of this effort came to be known as the BWW (Bunge-Wand-Weber) ontology. The authors then employed this theory to evaluate a number of conceptual modeling languages including NIAM (Weber & Zhang, 1991), ER (Wand et al., 1999), UML (Evermann & Weber, 2001) and OWL (Bera & Wand, 2004).

Despite being inspired by the goals of the BWW approach, we did not share their research direction. Bunge’s objective was doing philosophy of science and, being a physicist, chiefly, philosophy of the hard sciences. Conceptual Modeling, in contrast, is about “representing aspects of the physical and social world for the purpose of understanding and communication [and, hence] the contribution of a conceptual modeling notation rests in its ability to promote understanding about the depicted reality among human users” (Mylopoulos, 1992). As a result, any attempt to develop ontological foundations for conceptual modeling should take both human cognition and human linguistic competence seriously; it should be a project in descriptive metaphysics and not in revisionary ontology (Guarino & Guizzardi, 2006; Guizzardi, 2007b). This mismatch between the purposes for which Bunge’s ontology was developed and the requirements of ontological foundations for conceptual modeling became manifest in the literature as the predictions made by the BWW approach found themselves in strong contrast with the intuitions and practical knowledge of modelers and with some predictions unanimously shared by alternative approaches. For instance, around 2003, a number of authors showed that the BWW dictum that “mutual properties (or relations) should never be modeled as classes as they should not be allowed to have properties” was in conflict with modeling predictions made by other foundational theories (e.g., Jackendoff’s Semantic Structures (Veres & Hitchman, 2002; Veres & Mansson, 2005)) and with the intuition of modelers. Regarding the latter, as shown by (Hitchman, 2003), in recorded modeling sessions with practitioners, it became evident that reified relationships were frequently a part of these practitioners’ ontology.

It was clear to us from the outset that we needed an ontological theory that would countenance both individuals and universals and one that would include not only substantial individuals and universals but also accidents (particularized properties, moments, qualities, modes, tropes, abstract particulars, aspects, ways) and accident universals. In other words, we needed a Four-Category Ontology (Lowe, 2006). We needed particularized properties not only because they were of great importance in making sense of language and cognition (Davison, 2001; Parsons, 1990; Masolo et al., 2003) but because they
would repeatedly appear in the discourse of conceptual modelers. As previously mentioned, particularized relations (relationships) and particularized intrinsic properties (e.g., often represented by the so-called weak entities) are frequently modeled as bearers of other particularized properties in the practice of conceptual modeling. In fact, as demonstrated by (Guizzardi, 2005; Guizzardi et al., 2006; Guizzardi & Wagner, 2008; Guarino & Guizzardi, 2015), there are many conceptual modeling problems that can hardly be solved without considering particularized properties.

In initial papers for this project, we attempted to employ the GFO (General Formalized Ontology)/GOL (General Ontology Language) being developed in Leipzig, Germany, as our reference theory (Heller & Herre, 2004; Heller et al., 2004). More or less at the same time, a strong cooperation was established with the Laboratory of Applied Ontology (LOA) (Trento, Italy), which was during that period developing the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) (Masolo et al., 2003). In this setting, our first attempt was to unify DOLCE and GFO to produce a reference foundational ontology for conceptual modeling (hence, the name Unified Foundational Ontology) (Guizzardi & Wagner, 2004). Both theories were philosophically sound and formally characterized. Moreover, they were both based on the so-called Aristotelian Square, i.e., “Four-Category Ontologies”.

It became clear quite soon, however, that further investigation was needed, with a focus on the requirements of the conceptual modeling discipline. In particular, ontological foundations for conceptual modeling would demand micro-theories to address conceptual modeling’s most fundamental constructs, namely, Entity Types and Relationship Types (hence, the name of the so-called Entity-Relationship approach that gives the name to the most important conference in conceptual modeling!). So, any reference theory for conceptual modeling would need a rich theory of entity (object) types and a rich theory of domain (also called material) relations. In the case of the former, we needed something in the spirit of the ontology of universals underlying the OntoClean methodology (Guarino & Welty, 2009) in order to systematize a number of notions that were pervasive in the conceptual modeling literature (e.g., types, roles, phases or states, mixins) but for which there were no precise definitions or consensus (Guizzardi et al., 2004a). In that respect, GFO’s theory of universals still does not recognize these notions and DOLCE does not include universal as a category (DOLCE was designed as an ontology of particulars). Regarding the latter, DOLCE still does not include a theory of particularized relational properties (relational qualities) and the GFO theory of relations is subject to the so-called Bradley Regress (Bradley, 1893) and, hence, it can only be instantiated by infinite (logical) models. This feature makes it unsuitable for conceptual modeling applications. Finally, there were many additional specific aspects needed for a general ontology for conceptual modeling that were not addressed by the existing approaches (e.g., attributes and datatypes, different types of specialization relations between relations, a rich theory of cognitively/linguistically motivated part-whole relations). We needed to develop a full-blown foundational ontology specifically created to address these conceptual modeling requirements.

3. The Unified Foundational Ontology and its applications

The UFO ontology was developed by consistently putting together a number of theories originating from areas such as Formal Ontology in philosophy, cognitive science, linguistics and philosophical logics. It comprises a number of micro-theories addressing fundamental conceptual modeling notions. The ontology is divided in three strata dealing with different aspects of reality, namely:

- **UFO-A**: An Ontology of Endurants dealing with aspects of structural conceptual modeling. It is organized as a Four-Category ontology comprising theories of Types and Taxonomic Structures (Guizzardi et al., 2004a; Guizzardi, 2012) connected to a theory of object identifiers (including a formal semantics in a Sortal Quantified Modal Logics (Guizzardi, 2015)), Part-Whole Relations (Guizzardi, 2007a, 2009, 2010a, 2011), Particularized Intrinsic Properties, Attributes and Attribute Value Spaces (Guizzardi et al., 2004b, 2006; Guizzardi & Zamborlini, 2014) (including a theory of Datatypes as Measure Structures (Albuquerque & Guizzardi, 2013)), Particularized Relational Properties and Relations (Guizzardi & Wagner, 2008; Costal et al., 2011; Guarino & Guizzardi, 2015) and Roles (Guizzardi, 2006; Masolo et al., 2005);

- **UFO-B**: An Ontology of Perdurants (Events, Processes) dealing with aspects such as Perdurant Mereology, Temporal Ordering of Perdurants, Object Participation in Perdurants, Causation, Change and the connection between Perdurans and Endurants via Dispositions (Guizzardi et al., 2013a);
• **UFO-C**: An Ontology of Intentional and Social Entities, which is constructed on top of UFO-A and UFO-B, and which addresses notions such as Beliefs, Desires, Intentions, Goals, Actions, Commitments and Claims, Social Roles and Social Particularized Relational Complexes (Social Relators), among others (Guizzardi et al., 2008; Guizzardi & Guizzardi, 2010).

Over the years, UFO has been employed as a basis for analyzing, reengineering and integrating many modeling languages and standards in different domains (e.g., UML, TOGAF, ArchiMate, RM-ODP, TROPoS/i*, AORML, ARIS, BPMN) as well as for the development of Core and Domain Ontologies in different areas. For instance, it has been successfully used to provide conceptual clarification in complex domains such as Services (Nardi et al., 2015), Capabilities (Azevedo et al., 2015), Organizational Structures (Santos Junior et al., 2013), Communities (Almeida & Guizzardi, 2013), Goals and Motivations (Guizzardi et al., 2012, 2013; Azevedo et al., 2011), Constitutional Law (Griffo et al., 2015), Business Processes (Guizzardi & Wagner, 2011b; Santos Junior et al., 2010), Discrete Event Simulation (Guizzardi & Wagner, 2010, 2011a, 2012, 2013), Simulation for Land Covering and Use (Grueau, 2013), Software Quality (Shekhovtsov & Mayr, 2014), Software Measurement (Moretto & Barcellos, 2013), Foundations of Software Engineering (Henderson-Sellers, 2012; Henderson-Sellers et al., 2014), Software Requirements (Guizzardi et al., 2014), among many others. Of all applications of UFO, one deserves special attention, namely, the use of UFO-A in the design of an ontology-driven conceptual modeling language, which later came to be known as OntoUML. This language is discussed in the next section.

4. **OntoUML: Language, Engineering Support and its Applications**

OntoUML (Guizzardi, 2005) was conceived as an ontologically well-founded version of the UML 2.0 fragment of class diagrams. The idea was to employ the ontology-based language engineering method proposed by (Guizzardi, et al., 2005) (and later refined by (Guizzardi, 2013)) to design a language for structural conceptual modeling (i.e., for modeling aspects related to endurants) that would have two main characteristics. Firstly, the worldview embedded in the language through its conceptual primitives (what some authors would call the ontology behind the language (Guizzardi, 2007b)) should be isomorphic to the ontological distinctions put forth by UFO-A. Secondly, the language metamodel should incorporate formal syntactical constraints that would delimit the set of grammatically valid models of the language to those that represented intended state of affairs admitted by the underlying ontology. By doing this, we have managed to build a modeling language with explicitly defined formal and real-world semantics. This language serves as an engineering tool for enabling the use of formal ontological theories in the construction of conceptual models and domain ontologies.

Over the years, OntoUML has been adopted by many research, industrial and government institutions worldwide. In particular, it has been considered as a candidate for addressing the OMG SIMF (Semantic Information Model Federation) Request for Proposal, after a report of its successful use over the years by a branch of the U.S. Department of Defense (2011). In addition, some of the foundational theories underlying OntoUML have also been adopted by other conceptual modeling languages, e.g., ORM 2.0 (Halpin & Morgan, 2008; Halpin, 2010). It has been employed in a number of projects in different countries, in academic, government and industrial institutions, in domains such as Geology (Carbonera et al., 2015; Abel et al., 2015), Biodiversity Management (Albuquerque et al., 2015), Organ Donation (Pereira et al., 2015), Petroleum Reservoir Modeling (Werlang, 2015), Disaster Management (Moreira et al., 2015), Context Modeling (Brandt et al., 2014), Datawarehousing (Moreira et al., 2014), University Campus Management (Carolla & Spitta, 2014), Enterprise Architecture (Buckl et al., 2011), Metamodeling (Nijenhuis, 2011), Data Provenance (Serra, 2012), Measurement (Nunes et al., 2015), Logistics (Andreeva et al., 2015), Integration of Analytical Tools (Blackburn & Denno, 2015), Complex Media Management (Carolo & Burlamaqui, 2011), Telecommunications (Barcelos et al., 2011), Petroleum and Gas (Guizzardi et al., 2010), heart electrophysiology (Gonçalves et al., 2011), among many others.

The decision to build the language as a version of UML (technically, as a UML profile) was mainly motivated by the fact that UML (unlike many other conceptual modeling languages) has a standardized and explicitly defined metamodel and a significant set of metamodeling tools that can be used to manipulate and extend this metamodel. As a result, we managed to actually propose a concrete modeling language that could be used to create ontology-driven conceptual models and domain.

---

1. [http://www.omgwiki.org/architecture-ecosystem](http://www.omgwiki.org/architecture-ecosystem)
ontology in a variety of existing UML tools. For instance, as reported by (Guerson et al., 2015), an OntoUML plug-in has been implemented for the professional tool Enterprise Architect\(^2\). Moreover, as UML is a de facto modeling standard, we also managed to leverage existing knowledge and acceptance of the language (at least among computer scientists).

The first version of this adapted UML metamodel was proposed by (Guizzardi, 2005), fully implemented as a concrete UML profile by (Carraretto, 2010) and implemented as a model-based tool by (Benevides & Guizzardi, 2009). In that original OntoUML editor, we had support for model construction and formal verification (against formal metamodel constraints reflecting part of the UFO-A axiomatization). Over the years, a number of contributions have been aggregated in the proposal of a new editor that, besides supporting model construction and formal verification, incorporates support for a number of additional methodological features (Guerson et al., 2015):

(i) **A Pattern-Based approach for Model Construction:** As demonstrated by (Guizzardi et al., 2011; Guizzardi, 2014), OntoUML is actually a Pattern-Based Language in the sense that its modeling primitives are patterns, i.e., higher-granularity clusters of modeling elements that can appear in a model only in particular fixed configurations. Moreover, these patterns are of an ontological nature, as they directly reflect the ontological micro-theories underlaying the language. Examples include the role with disjoint allowed types modeling pattern proposed by (Guizzardi et al., 2004a), the relator-material relation pattern (Guizzardi & Wagner, 2008), the qualities with alternative quality spaces pattern (Guizzardi et al., 2006), among many others. Furthermore, the editor also implements a number of topological patterns that allows for isolating the scope of transitivity of part-whole relations (Guizzardi, 2009). Finally, the editor allows for the extraction of Domain-Related Patterns from Core Ontologies (e.g., the Employment, Organization Structure and Project Involvement patterns exemplified by (Ruy et al., 2015)) such that these patterns can be reused for the construction of Domain Ontologies;

(ii) **Model Verbalization:** Model verbalization stands for the activity of generating a rendering of the model in (controlled) natural language. This process is very useful, for example, to allow domain experts that are not well-versed in the modeling language’s notation, to access a partial view of what is represented in a conceptual model. The editor incorporates a functionality for automatically generating model verbalization in structured English following a slightly modified version of the SBVR (Semantics for Business Vocabularies and Rules) OMG proposal\(^3\);

(iii) **Support for the Representation of Domain-Specific Formal Constraints:** In order to cover domain constraints that cannot be represented using the language’s diagrammatic notation, the current editor supports the specification of OCL (Guerson et al., 2014) and temporal OCL formal constraints (Guerson & Almeida, 2015). The editor provides support for syntax highlighting, code-completion and syntax verification (parsing) for textual constraints;

(iv) **Model Validation:** This approach addresses conceptual model validation by using visual simulation. In particular, in the strategy proposed by (Benevides et al., 2010; Braga et al., 2010) and implemented in this later version of the editor, we have the automatic generation of visual instances (exemplars) of a given conceptual model such that the modeler can be confronted with what her model is actually representing. In other words, the strategy is to systematically contrast the set of formally-valid instances of a given conceptual model (automatically generated by the visual simulator) with the set of intended instances of that model (i.e., instances that represent admissible state of affairs in reality), which exists only in the modeler’s mind. Once the modeler detects a deviation between valid and intended instances (either due to overconstraining or underconstraining of the model), she rectifies the model, for instance, by the inclusion of formal domain-specific constraints;

\(^1\)http://www.sparxsystems.com.au/

\(^2\)http://www.omg.org/spec/SBVR/1.2/
(v) **Ontology Codification:** In the ontology engineering process defended by (Guizzardi, 2007b; Guizzardi & Halpin, 2008; Guizzardi, 2010b), we have advocated a separation between the modeling approaches that should be used for the conceptual modeling phase of ontology engineering and the representation approaches that could be used to realize alternative implementations of a given ontology (as a conceptual model). For instance, given a conceptual model representing a domain ontology in OntoUML, we can have different mappings to different codification languages (e.g., OWL DL, RDF, F-Logic, Haskell, Relational Database languages, CASL, among many others). The choice of each of these languages should be made to favor a specific set of non-functional requirements. Moreover, within the solution space defined by these codification languages, we have a multitude of choices regarding, for instance, decidability, completeness, computational complexity, reasoning paradigm (e.g., closed versus open world, adoption of a unique name assumption or not), expressivity (e.g., regarding the need for representing modal constraints, higher-order types, relations of a higher arity), verification of finite satisfiability, among many others. In the current version of the editor, we have implemented six different automatic mappings from OntoUML to OWL contemplating different transformation styles that were designed to address different sets of non-functional requirements (Guizzardi & Zamborlini, 2014, Zamborlini & Guizzardi, 2010). It is important to highlight that other authors have implemented alternative mappings from OntoUML to languages such as XML (Baumann, 2009), Smalltalk (Pergl et al., 2013), OWL (Barcelos et al., 2013) and a version of a Modal Prolog (Araujo, 2015);

(vi) **Anti-Pattern Detection and Rectification:** Given the diffusion of the language, we have managed to assemble a model repository containing OntoUML models in different domains (e.g., telecommunications, government, biodiversity, bioinformatics), different sizes (e.g., ranging from dozens of concepts to thousand of concepts), and produced in different types of contexts (e.g., ranging from academic exercises of novices to models produced by teams of practitioners in industrial or government settings) (Sales & Guizzardi, 2015). By using this model repository as a benchmark, in three different empirical studies, (Guizzardi & Sales, 2014; Sales & Guizzardi, 2015) managed to show that this approach for model validation via visual simulation is not only able to detect deviations between formally valid model instances and intended model instances, but is also able to detect recurrent structures that tend to cause these deviations, i.e., **ontological anti-patterns.** Once these anti-patterns are catalogued, they were able to devise solution patterns, i.e., solution proposals that eliminate the deviation between valid instances and intended instances (Guizzardi & Sales, 2014; Sales & Guizzardi, 2015). The current version of the editor implements both a mechanism for anti-pattern detection and an implementation of these proposed rectification solutions. Sales and Guizzardi (2015) have presented a validation study developed with a large industrial model and managed to empirically demonstrate, for the vast majority of the identified anti-patterns, a very high correlation between the presence of these anti-patterns and the adoption of our proposed solutions for ontology-based model enhancement.

5. Learning from “Systematic Language Subversions” to evolve UFO and OntoUML

The observation of the application of OntoUML over the years conducted by a variety of groups in a variety of domains also amounted to a very fruitful empirical source of knowledge for us regarding the language and its foundations. In particular, we have managed to observe a number of different ways in which people would slightly **subvert** the syntax of the language, ultimately creating what we could call **“systematic subversions”** of the language. These “subversions” would (purposefully) produce models that were grammatically incorrect, but which were needed to express the intended characterization of their underlying conceptualizations that could not be expressed otherwise. Loosely speaking, following the semiotic engineering principle of seeing representations as dialogues between designer and user (de Souza, 2005), we had the feeling that the users of the language were “speaking to us” via these “systematic subversions”.

One example is the representation of events in OntoUML models. As previously mentioned, the language was created to represent structural conceptual models expressing endurantistic aspects of reality. However, systematically, a number of authors, in different institutions and in different domains (e.g., Moretto & Barcellos, 2013; U.S. Department of Defense, 2011) started to produce OntoUML models in which perdurants (and perdurant relations) would appear as modeling primitives. These
examples were concrete cases of the need to also represent perdurants in structural conceptual models (Olivé, 2007). In order to respond to these modeling requirements, we intend to, once more, employ the aforementioned ontology-based language engineering process, extending the metamodel the language to incorporate the ontological distinctions and axiomatization prescribed by UFO-B (Guizzardi et al., 2013a). Doing that, however, requires that a number of new research problems be addressed. To cite one of these challenges, given that the introduction of this new perspective substantially increases the complexity of the resulting diagrams, new complexity management theories and tools for OntoUML diagrams need to be developed (e.g., dealing with model filtering, modularization, viewpoint selection).

A second case systematic “language subversions” led us to reconsider some of the theoretical foundations underlying the language, i.e., it led us to rethink and evolve a core theory in UFO. As previously mentioned, in the original version of UFO (and, hence, also in OntoUML), we had a number of ontological distinctions representing different sorts of universals. These include distinctions between substance sortals (kinds), phased-sortals (roles and phases) and non-sortals (categories, mixins and role mixins). These distinctions, however, were considered to be distinctions among object universals. However, consciously ignoring this rule, users of the language started to systematically employ these distinctions also when characterizing universals whose instances are existentially dependent endurants (i.e., moments/aspects/particularized properties such as modes and relators). These “subversions” called our attention to the fact that, like full-fledged endurants, modes and relators also have their identity supplied by substance sortals (i.e., mode kinds and relator kinds) and are also subject of both essential and accidental properties. If these endurants can be subjects of modal properties, then they can also contingently instantiate anti-rigid types. In other words, phases, roles, roleMixins, mixins, categories, etc., (as sorts of types characterized by formal ontological meta-properties) could also be used to characterize mode and relator universals. This realization triggered a process that led to a much richer theory of material relations and particularized relational properties in UFO, which in turn triggered an evolution of the OntoUML model, with direct interesting consequences to the practice of conceptual modeling (Guarino & Guizzardi, 2015). In fact, in an upcoming version of UFO, these different sorts of universals will be applicable to all categories of endurant universals, including the so-called powertypes. Powertypes can be loosely defined as “types whose instances are other types” (e.g., the type Organization Position, which can be instantiated by the types Director, Manager and Secretary, or the type Bird Species, which can be instantiated by the types Golden Eagle and Emperor Penguin). (Guizzardi et al., 2015) defend that powertypes should not be interpreted as higher-order universals but as concrete resemblance structures (Armstrong, 1990) that are themselves endurants.

With the increasing popularization of OntoUML and its associated tools, this dialogue with language users will continue, with more implications to be drawn to OntoUML and UFO from their practical applications in various domains. In this way, the evolution of the language and its conceptual foundations can remain anchored in the conceptual modeling discipline, not only taking the real-world semantics of conceptual modeling primitives seriously, but also addressing recurrent conceptual modeling problems against the solid background of formal ontology. We believe this combined use of insights from theory and practice is key to applied ontology in general, but is essential to ontology-driven conceptual modeling.

Acknowledgements. The authors are grateful to Ricardo Falbo, Monalessa Barcellos, Nicola Guarino, John Mylopoulos, Heinrich Herre and the members of NEMO for many years of fruitful collaboration. Finally, we would like to express our gratitude to Terry Hapin, Alex Borgida and Yair Wand for their valuable comments to a previous version of this paper.

References


Ontologies in Medicine, University of Leipzig.


87. 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.


