A Semantic Foundation for Organizational Structures: A Multi-Level Approach

Conference Paper · September 2015

DOI: 10.1109/EDOC.2015.18

2 authors:

Victorio Albani Carvalho
Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo…

João Paulo A. Almeida
Universidade Federal do Espírito Santo

A Semantic Foundation for Organizational Structures: A Multi-Level Approach

Victorio Albani Carvalho [1,2], João Paulo A. Almeida[1]
[1]Ontology & Conceptual Modeling Research Group(NEMO)
Federal University of Espírito Santo (UFES), Vitória (ES), Brazil
[2]Research Group in Applied Informatics, Informatics Department
Federal Institute of Espírito Santo (IFES), Colatina (ES), Brazil
victorio@ifes.edu.br; jpalmeida@ieee.org

***Abstract*—Conceptualizing the organizational structure domain requires considering multiple levels of classification, with both *types* and *types of types* included in the domain of enquiry (e.g., types of organizational units and particular organizational units). In this paper we propose a semantic foundation for the organizational structure domain that is capable to address the multi-level modeling issues. We present a core organizational structure ontology built with the combination of a foundational ontology (UFO) and a multi-level modeling theory (MLT). This ontology serves to provide semantic foundations for enterprise modeling languages but also as a basis for the development of enterprise-specific ontologies. We discuss our contributions with respect to existing multi-level modeling approaches and with respect to a number of prominent enterprise modeling frameworks, languages and enterprise ontologies.**

***Keywords*—organizational structure; enterprise modelling; ontology; multi-level modeling.**

I. INTRODUCTION

Enterprise Architecture (EA) frameworks typically consider an organization as a system whose elements include: (i) organizational activities structured in business processes and services; (ii) information systems supporting organizational activities; (iii) underlying information technology (IT) infrastructures, and (iv) organizational structures. This last domain focuses on the business agents that perform tasks and seek to achieve goals, encompassing the definition of business roles, organizational relationships, organizational units, work groups, etc. The relevance of organizational structure is clear from a management perspective in that it defines authority and responsibility relations between the various elements of an enterprise. Further, from the perspective of enterprise information systems, organizational actors can be considered as system owners, system maintainers, system users or simply system stakeholders in general, affecting the usage and evolution of such systems [1]. Given the importance of the organizational structure domain, the support for modeling constructs for organizational structure is present in a number of enterprise modeling frameworks and languages including ARIS [2], ArchiMate [3], DoDAF [4], and RM-ODP [5], and also in a number of ontologies for the organizational structure domain, such as the AIAI Enterprise Ontology [6], the TOVE Enterprise Ontology [7], and the W3C Org Ontology [8].

One of the key challenges in conceptualizing the organizational structure domain is that it can span multiple levels of classification, with *types* and *types of types* being part of the domain of enquiry (which has been referred to in the literature as *multi-level modeling* [9] [10]). For instance, organizations may be staffed according to *role types* such as “Professor”, “Dean”, “Secretary”, “Project Leader”. They may also be structured according to different *types of organizational units* such as e.g., “Division”, “Department”, “Section”, each of which may impose constraints on some required *role types* (e.g. each “Department” of the “Federal University of Espírito Santo” has a “Dean”). Thus, to describe the conceptualization underlying this domain, one needs to represent entities of different (but nonetheless related) classification levels, such as *individual persons* (“John”, “Mary”), *role types* (“Dean”, “Secretary”), *organizational units* (“Sales Division of Coca-Cola Inc.”, “Computer Science Department of the Fed. Univ. of Espírito Santo”) and *organizational unit types* (“Department”, “Division”). Furthermore, there is a large diversity of organizational structuring approaches in different enterprise settings, making the enumeration of a fixed set of role types and organizational unit types untenable.

Some of the approaches we have investigated for the representation of the organizational structure domain, such as, e.g., ArchiMate [3] and the W3C Org Ontology [8] do not offer modeling constructs to represent types of organizations and organizational units, focusing only on offering constructs for users to capture instance-level notions. Some other approaches such as ARIS do cover types of roles and organizational units (including elements such as “Position Type”, “Organization Unit Type” and “Person Type”), but present several semantic ambiguities [1], lacking a clear semantic foundation for the concepts in the organizational structure domain (a problem which affects many other approaches concerning role-related concepts as discussed in [11]).

Considering that the main goal of enterprise models is to represent organizational reality faithfully and thus serve for the purposes of documentation, analysis and communication, EA modeling languages could benefit from the use of well-founded conceptual models as theoretical basis. The semantic shortcomings of some enterprise modeling approaches have motivated several efforts in the past decade into suitable conceptual foundations to inform the design or redesign of enterprise modeling approaches. For example, the Unified Foundational Ontology (UFO) [12] has been used to identify issues in the modeling of roles in some enterprise modeling approaches [11] and other related organizational structure elements of ARIS [1]. That has led to a number of proposals of improvements to these EA approaches. More recently, more specialized semantic foundations have been explored, employing not only a foundational ontology but also core organizational ontologies in the evaluation and redesign of modeling languages. For example, UFO-S [13] has been used to evaluate the service constructs of ArchiMate [14] and the O3 Organizational Ontology (aligned with the foundational ontology) in order to inform the extension of the ArchiMate metamodel for organizational structure modeling [15]. This allows for a more detailed analysis of certain domain-specific language aspects, incorporating relevant notions from the specialized literature for a particular domain of enquiry.

Despite these advances, a semantic foundation for the organizational structure domain that is capable of addressing the multi-level modeling issues is still lacking. In this paper we address this gap by presenting a core organizational structure ontology built with the combination of a foundational ontology (UFO) and a multi-level modeling theory, called MLT [16]. This core ontology serves as a semantic foundation for EA modeling languages and also as a basis for the development of enterprise-specific ontologies.

This paper is further structured as follows: section II presents an overview of our approach, which defines a hierarchy of conceptual models at different levels of specificity; section III presents the MLT multi-level theory and discusses its combination with UFO to support the definition of ontologically well-founded multi-level conceptual models; section IV presents the core organizational structure ontology built with the UFO-MLT combination; section V discusses related work and section VI presents concluding remarks.

II. The Modeling Approach

In order to address the challenge of multi-level modeling, we have proposed in [16] an axiomatic theory called MLT. Similarly to [17] it admits the existence of entities, which are, simultaneously, type (class) and instance. MLT precisely defines the relations that may occur between elements of different classification levels, combining the solutions based on "clabjects" with solutions based on power types [18] [19].

To support the construction of ontologically well-founded multi-level conceptual models, we employ a combination of UFO [12] and MLT [16]. The Unified Foundational Ontology (UFO) is a domain independent system of categories aggregating results from disciplines such as Analytical Philosophy, Cognitive Science, Philosophical Logics and Linguistics. Over the years, UFO has been successfully employed to analyze the classical conceptual modeling constructs including Object Types and Taxonomic Structures, Part-Whole Relations, Intrinsic and Relational Properties, Weak Entities, Attributes and Datatypes [12], and also organizational structure fragments of enterprise modeling approaches [1], [20].

MLT is used as the topmost layer of a hierarchy of conceptual models (see Fig. 1). The concepts and patterns of MLT are applied to establish the relation between MLT and UFO, and later to establish the relation between a domain conceptual model and UFO-MLT. More specifically, the concepts of UFO instantiate and specialize elements of MLT, thereby respecting MLT's axioms and leveraging the use of structural relations and patterns of MLT in UFO. In turn, the concepts of domain conceptual models instantiate and specialize concepts of MLT and UFO, respecting all rules and patterns of both MLT and UFO. Conceptual models built with the UFO-MLT combination benefit from the ontological distinctions of UFO as well as MLT's concepts and patterns for multi-level modeling [21].

In this paper, we adopt this hierarchical approach to build a core organizational structure ontology applying both UFO and MLT concepts. Following such approach, the core organizational structure ontology may be extended by enterprise-specific ontologies that instantiate and specialize the core ontology distinctions with the concepts that are required in a particular organizational setting. Such a hierarchical approach is required to cope with the large diversity in organizational structures and structuring approaches. The core ontology defines generic concepts such as "Organization Type", "Unit Type" and "Business Roles" and refrains from enumerating specific unit types such as "Division" and "Department", as well as specific business roles such as "Vendor", "Manager" and so on. These variations will be accommodated at more specific ontologies that extend the core organizational ontology to provide the elements that are used to account for organizational reality in particular enterprise contexts. An example of a hierarchy of models is illustrated in Fig. 1. MLT serves as the topmost layer with UFO immediately below. The core organizational structure ontology extends UFO-MLT. Then, two domain-specific organizational ontologies extend the core ontology: one focused on universities and another for manufacturing companies. Finally, the domain-specific ontologies are further refined to provide concepts required in the context of specific universities (e.g., the Federal University of Espírito Santo, UFES) and specific manufacturing companies (such as the Ford Motor Company).

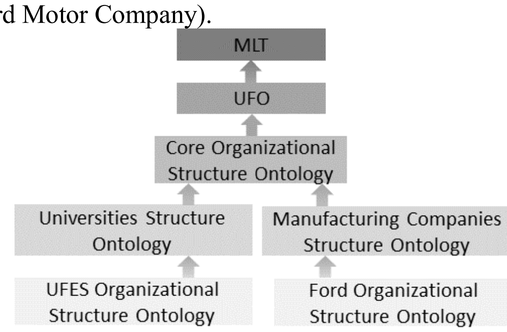

Fig. 1. Illustrating the hierarchy of models.

III. Foundations: MLT and UFO

A. The MLT Multi-Level Theory

MLT [16] distinguishes between types and individuals admitting types having individuals as instances as well as types that have other types as instances. In order to accommodate these varieties of types, the notion of type order is used in MLT. Types having individuals as instances are called first-order types, types whose instances are first-order types are called second-order types and so on.

In order to link types to the entities that fall under such types, MLT defines a primitive *instance of* relation. This relation is represented by a ternary predicate *iof(e,t,w)* that holds if an entity *e* is *instance of* an entity *t* (denoting a type) in a world *w*. Indexing the instantiation relation to possible worlds allows MLT to support dynamic classification, admitting thus types that apply contingently to their instances (e.g., John is an instance of student in w but not in w', when he has graduated.)

We build up the axiomatic theory defining the conditions for entities to be considered individuals, using the logic constant "Individual". Thus, an entity is an instance of "Individual" iff it cannot possibly be related to another entity through instantiation. The constant "First-Order Type" (or shortly "1stOT") characterizes the type that applies to all entities whose instances are instances of "Individual". Analogously, each entity whose possible extension contains exclusively instances of "1stOT" is an instance of "Second-Order Type" (or shortly "2ndOT"). It follows from this definition that "Individual" is instance of "1stOT" which, in turn, is instance of "2ndOT". We call "Individual", "1stOT" and "2ndOT" the basic types of MLT. According to MLT, every possible entity must be instance of exactly one of its basic types (except the topmost type). For our purposes in this paper, first- and second-order types are enough. However, this scheme can be extended to consider as many orders as necessary.

Fig. 2 illustrates the elements that form the basis for MLT, using a notation that is inspired in UML. We use the UML class notation to represent types. We use associations as usual to represent relations between instances of the related types (predicates that may be applied to instances of the related types). Since UML does not allow for the representation of links between classes, we use dashed arrows to represent relations that hold between the types, with labels to denote the names of the predicates that apply. This notation is used in all further diagrams in this paper. It is important to highlight here that our focus is not on the syntax of a multi-level modeling language and we use these diagrams to illustrate the concepts. A complete formalization of MLT can be found in [16].

Fig. 2. Basic foundations of MLT: basic types and instance of relation.

Some structural relations to support conceptual modeling are defined in MLT, starting with the ordinary *specialization* between types. A type *t specializes* another type *t'* iff in all possible worlds all *instances of t are also instances of t'*. According to this definition every type specializes itself. Since this may be undesired in some contexts, we define the *proper specialization* relation as follows*: t proper specializes t' iff t specializes t' and t is different from t'*. Note that the definitions presented thus far guarantee that both specializations and proper specializations may only hold between types of the same order (see the upper part of Fig. 3).

Every type that is not one of MLT's basic types (e.g., a domain type) is an instance of one of the basic higher-order types (e.g., "1stOT", "2ndOT"), and, at the same time proper specializes the basic type at the immediately lower level (respectively, "Individual" and "1stOT"). Fig. 3 illustrates this basic pattern of MLT. Since "Person" applies to individuals, it is instance of "1stOT" and proper specializes "Individual". The instances of "Person Age Phase" are specializations of "Person" (e.g. "Child" and "Adult"). Thus, "Person Age Phase" is instance of "2ndOT" and proper specializes "1stOT". This pattern of MLT is used repeatedly in our hierarchical approach, and is the basis to establish a clear relation between levels of specificity in the multi-level approach.

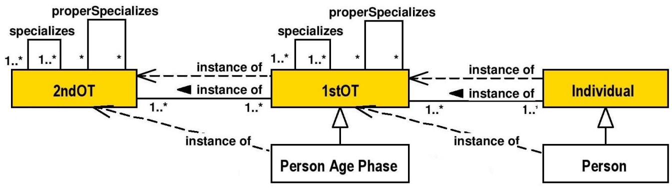

Fig. 3. Illustrating the basic pattern of MLT: "Person" is instance of "1stOT"and specializes "Individual" while "Person Age Phase" is instance of "2ndOT"specializing "1stOT".

In addition to the relations that occur between entities of the same order, MLT defines *cross-level structural relations* between types of adjacent orders. These relations support an analysis of the notions of power type in the literature, leading to their incorporation in the theory.

First, based on the notion of power type proposed by Cardelli [18], MLT defines that a *power type* relation between a higher-order type and a base type at an order lower: *a type t is power type of a base type t' iff all instances of t specialize t' and all possible specializations of t' are instances of t*. For example, consider a type called "Person Powertype" such that all possible specializations of "Person" are instances of it and, conversely, all its instances specialize "Person". In this case, "Person Powertype" is the power type of "Person". Since "Person" is instance of "1stOT", "Person Powertype" is instance of "2ndOT" and specializes "1stOT" (see Fig. 4). Note that it follows from the definition of power type that "*1stOT" is power type of "Individual"*. Analogously, *"2ndOT" is power type of "1stOT"*.

Second, based on Odell's definition for power types [19], MLT defines the *characterization* relation between types of adjacent levels: *a type t characterizes a type t' iff all instances of t are proper specializations of t'*. Note that there may be specializations of the base type t' that are not instances of t. For instance in Fig. 4, "Person Role" (with instances "Manager" and "Researcher") characterizes "Person", but is not a power type of "Person", since there are specializations of "Person" that are not instances of "Person Role" ("Child" and "Adult").

We define some variations of characterization, which are useful to capture further constraints in multi-level models. We consider that *a type t completely characterizes t' iff t characterizes t' and every instance of t' is instance of, at least, an instance of t*. Moreover, *iff t characterizes t' and every instance of t' is instance of, at most, one instance of t it is said that t disjointly characterizes t'*. Finally, a common use for the notion of power type in literature considers a second-order type that, simultaneously, completely and disjointly characterizes a first-order type. To capture this notion we defined the *partitions* relation. Thus, *t partitions t' iff each instance of t is instance of exactly one instance of the base type t'*. For example of the partitioning relation, consider the second-order type called "Person Age Phase" with instances "Child" and "Adult" (Fig. 4). (In UML this kind of constraint is represented through a generalization set, see [16] for a detailed comparison).

B. Combining MLT and The Unified Foundational Ontology

UFO includes a taxonomy of individuals and a taxonomy of universals (more specifically *first-order* universals). The notion of individual of UFO and MLT are coincident and the UFO notion of universal is encompassed by the MLT notion of first-order type ("1stOT"). Therefore, applying the MLT basic

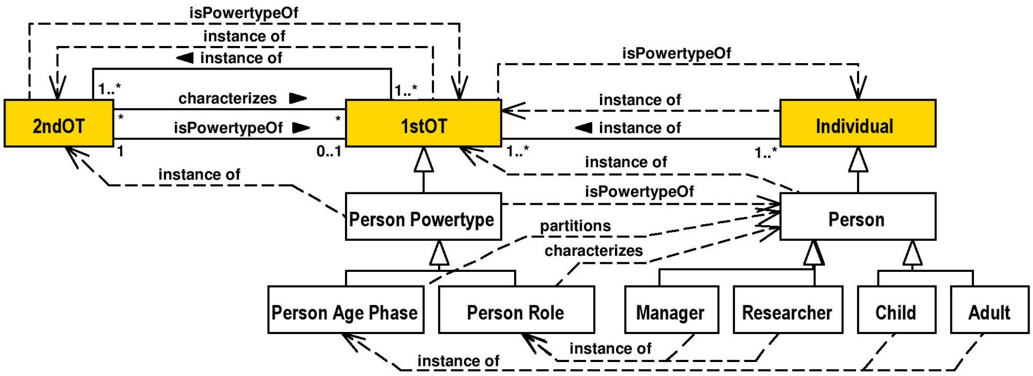

Fig. 4. MLT Cross-level relations: "Person Powertype" is power type of "Person", thus all specializations of "Person" are instances of "Person Powertype"; "Person Age Phase" partitions "Person" having "Child" and "Adult" as instances; "Person Age Phase" characterizes "Person" having "Manager" and "Researcher" as instances.

pattern to UFO taxonomies, it follows that the concepts in UFO's taxonomy of individuals are instances of "1stOT" specializing "Individual" while the concepts in the taxonomy of universals are instances of "2ndOT" specializing "1stOT" (see Fig. 5).

The topmost distinction in the taxonomy of individuals is that between endurants and events. Endurants (as opposed to events) are the individuals said to be wholly present whenever they are present, i.e., they can endure in time, suffering a number of qualitative changes while maintaining their identity (e.g., a house, a person). Since in this paper we are especially interested in a portion of UFO that accounts for structural (as opposed to dynamic) aspects of conceptual modeling, we focus solely on endurants. Endurants are further classified into Substantials and Moments. Substantials are existentially-independent endurants (e.g. a person, a car). A moment, in contrast, is an endurant that inheres in, and, therefore, is existentially dependent of, another endurant(s). Moments that are dependent of one single individual are Intrinsic Moments (e.g. a person's age) whereas moments that depend on a plurality of individuals are instances of Relator (e.g. a marriage, an employment, an enrollment).

These distinctions among individuals are reflected in the taxonomy of universals. Instances of "Intrinsic Moment Universal" apply to intrinsic moments (e.g. "Age"), instances of "Relator Universal" have relators as instances (e.g. "Marriage") and instances of "Substantial Universal" have substantials as instances (e.g. "Person"), i.e. instances of the entities in the taxonomy of universals specialize the corresponding entities in the taxonomy of individuals. Thus, following MLT, "Intrinsic Moment Universal" *characterizes* "Intrinsic Moment", "Relator Universal" *characterizes* "Relator" and "Substantial Universal" *characterizes* "Substantial" (see Fig. 5).

The ontological category of "Substantial Universal" is further specialized according to the ontological notions of identity and rigidity. Substantial universals that carry a uniform principle of identity for their individuals are instances of "Sortal Universal" (e.g., "Person", "Car", "Organization"). In contrast, instances of "Non Sortal Universal" represent an abstraction of properties that are common to instances of various sortals (e.g., the non-sortal "Insurable Item" describes properties that are common to entities of different sortals such as "House", "Car", "Work of Art"). Moreover, a universal is said to be rigid if it classifies its instances necessarily (in the modal sense). In other words*, if a universal T is rigid, then an instance x of T cannot cease to be an instance of T without ceasing to exist* (e.g., "Person", "Organization"). In contrast, a universal is anti-rigid if its instances can move in and out of the extension of that universal without ceasing to exist (e.g., "Student", "Employee"). Rigid sortals that provide a uniform principle of

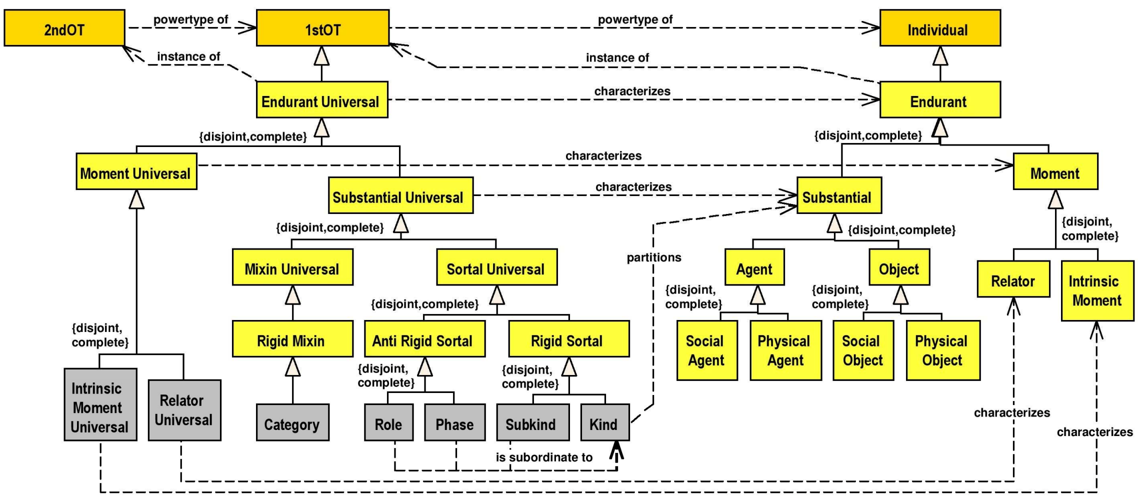

Fig. 5. Applying MLT to UFO: every type in UFO`s taxonomy of individuals are instances of "1stOT" specializing "Individual" while the types in the taxonomy of universals are instances of "2ndOT" specializing "1stOT" (we have represented the instantiation relations only to the topmost type of each UFO taxonomy).

identity to their instances are termed a Kind (e.g "Person"). Instances of "Kind" may be specialized in other rigid sortals, which are instances of a Subkind (e.g. "Man"). Anti-rigid sortals are further classified into the categories Role or Phase. Instances of "Role" classify substantials through the relational properties they bear in the scope of a relational context (e.g. "Employee", "Husband", "Student") whereas instances of Phase define partitions of a Kind depending on one or more of its intrinsic properties (e.g "Child", "Living Person"). Rigid non-sortals that represent abstractions of properties that apply to instances of different kinds are called Category universals (e.g., "Legal Entity" abstracting persons and organizations).

Since each instance of "Substantial" is an instance of exactly one instance of "Kind" (the kind that supplies the principle of identity), following MLT, "Kind" *partitions* "Substantial". In addition, since they carry (but do not supply) a principle of identity, instances of "Subkind", "Phase" and "Role" must specialize an instance of "Kind" that supplies such principle. Thus, in MLT terms, "Subkind", "Phase" and "Role" are *subordinate to* "Kind". Subordination between two higher-order types implies specializations between their instances i.e., *t is subordinate to t' iff every instance of t proper specializes an instance of t'*.

The UFO taxonomy of individuals includes a social layer that specializes its core with distinctions to account for intentionality and social reality [22]. It distinguishes between agentive and non-agentive objects. Agentive objects (instances of "Agent") can perform actions and have intentional moments (intentions, desires and beliefs). Agents are differentiated in "Physical Agents" (e.g., a person) and "Social Agents" (e.g., an organization). The latter are created by speech acts and normative descriptions recognized by a numbers of agents.

IV. A Core Organization Ontology in UFO-MLT

In order to benefit from the ontological distinctions of UFO as well as the basic concepts and patterns for multi-level modeling of MLT, our organization ontology adheres to the rules of both theories. Thus, every domain *first-order* type of our ontology (i) instantiates one of the leaf ontological categories of UFO's taxonomy of universals (the types in dark grey in Fig. 5) and, consequently, instantiates MLT's "1stOT"; and (ii) simultaneously, specializes one of the leaf ontological categories of UFO's taxonomy of individuals and thus, specializes "Individual". Further, every *second-order type* of our ontology specializes one of the leaf ontological categories of UFO's taxonomy of universals and has an *MLT cross-level relation* with a *first-order type*, following the patterns proposed in [21].

This ontology has been defined as a simplified version of O3 [23] [15] focusing on the general concepts and highlighting the aspects that are specific to multi-level modeling. It adds to O3 *second-order types* to map the notions of "Unit Type", "Organization Type" and "Assignment Type". We discuss the ontology following two points of view: (i) organizational structure (in subsection A) and (ii) organizations roles and allocations (in subsection B).

A. Organizational Structure

Fig. 6 illustrates the fragment of the ontology related with organizational structure concepts (types shaded in dark gray). The topmost concept is "Organization", specializing the UFO notion of "Social Agent". As defined in [24], organizations are (artificial) social units built with the explicit intention of pursuing specific goals. Organizations include corporations, armies, hospitals and churches, but exclude tribes, ethnic groups, families and groups of friends. Members of an organization (which constitute the organization at a particular point in time) can be replaced or relocated to other functions while the organization persists in time.

We specialize "Organization" into "Formal Organization" and "Organizational Unit". Formal organizations are formally recognized by the external environment. Their creation is determined by normative descriptions or speech acts which are recognized by the normative context in which they exist. Examples of formal organization include "Microsoft Inc.", "The UK Government" and the "Federal University of Espírito Santo". Formal organizations may be composed of other formal organizations and of organizational units (see [20] for a discussion on the whole-part relation of UFO applied at the organizational context.).

Organizational units are those organizations that are only recognized in the internal context of a formal organization and represent the working groups of a formal organization. Examples of organizational units include the Marketing Department of Ford and the Sales Division of Coca-Cola. Since we consider that "Formal Organization" and "Organizational Unit" provide a principle of identity to their instances, both are considered instances of the UFO notion of "Kind". The more general notion of "Organization" is an instance of the UFO notion of "Category".

Specializations of "Formal Organization" are instances of the higher-order "Formal Organization Type". Examples of formal organization types specializing "Formal Organization" include "University", "Corporation", "Government", and "Hospital". Given that "Formal Organization" is considered an instance of "Kind", it follows that instances of "Formal Organization Type" are subkinds in UFO sense and, thus, "Formal Organization Type" specializes the UFO notion of "Subkind". Further, since all instances of "Formal Organization" are also instances of at least one instance of "Formal Organization Type" we can conclude that "Formal Organization Type" *completely characterizes* (in MLT sense) "Formal Organization". Analogously, specializations of "Organizational Unit", such as "Department", "Division" and "Project" are instances of "Organizational Unit Type". Thus, "Organizational Unit Type" *specializes* "Subkind" and *completely characterizes* "Organizational Unit".

A formal organization type may define the possible structures of its instances by constraining the types of organization (organizational units or other formal organizations) that may compose organizations of such type. Analogously, an organizational unit type may specify that an instance of it must be composed of other units of specific types. This is the basis for the definition of domain-specific types in an ontology that extends the core ontology. For instance, consider a domain ontology about university organizational structure. In this context, we can define "University" as an instance of "Formal Organization Type" (and thus, as a specialization of "Formal Organization"). Considering that universities are structured into faculties, which, in turn, are organized into departments we can define both "Faculty" and "Department" as instances of "Orga-

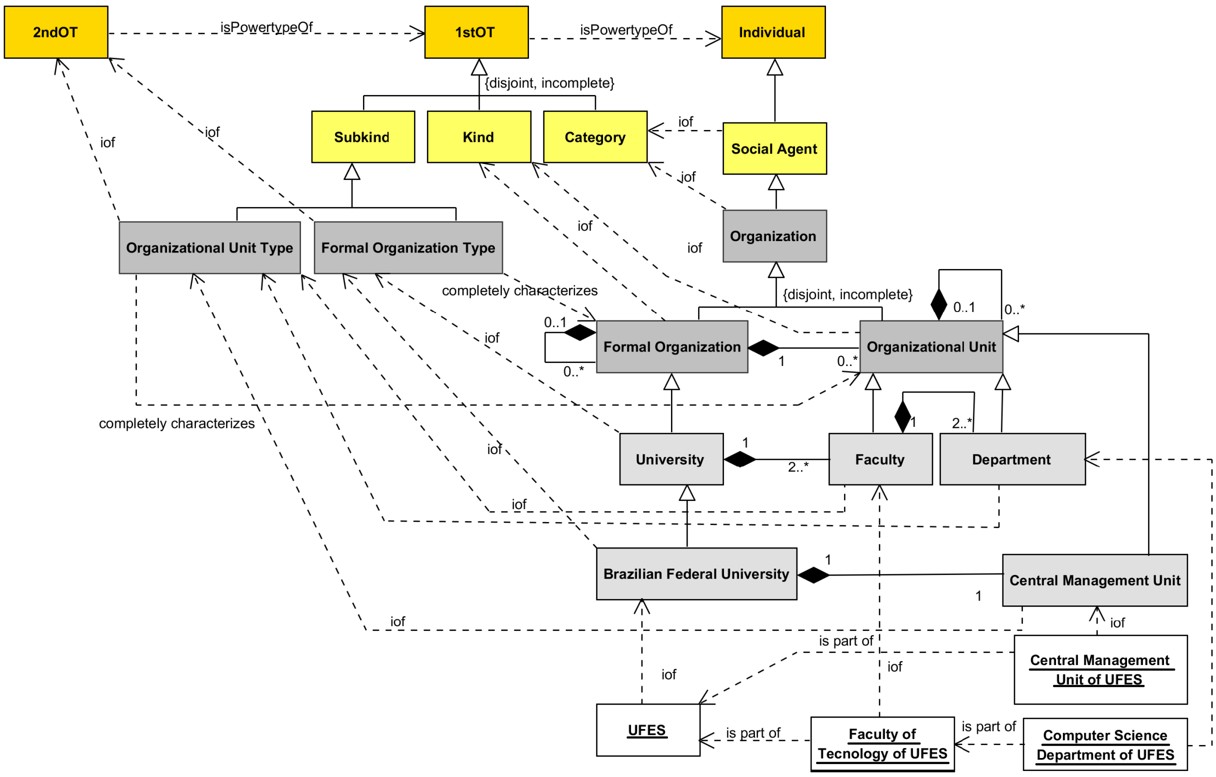

Fig. 6. Illustrating the definitions of the core organizational structure ontology and its use as the basis of a hierarchy of models: types that are part of the core ontology are shaded in dark gray; the types in light gray compose a fragment of a domain ontology about universities organizational structure; finally the types in white compose a model representing part of the organizational structure of the Federal University of Espírito Santo.

nizational Unit Type" and, thus, as specializations of "Organizational Unit". Further, we may state that a "University" is composed of at least two faculties and each "Faculty" of a "University" is mandatorily composed of at least two "Departments". Fig. 6 illustrates it representing these domain-specific concepts in light gray.

Both formal organization types and organizational unit types may be specialized into domain-specific types. For example, supposing that all Brazilian federal universities must comply with the previous definition of university and, additionally, must have a "Central Management Unit", we may capture it by creating: (i) an instance of "Formal Organization Type" called "Brazilian Federal University" as a specialization of "University", and (ii) an instance of "Organizational Unit Type" called "Central Management Unit" having a mandatory composition relation with "Brazilian Federal University". Considering these two new types are part of the universities ontology, they are depicted in light gray in Fig. 6.

This hierarchy of models can be further extended by creating models to express the structure of specific universities, such as the Federal University of Espírito Santo (UFES). For example, "UFES" can be represented as an instance of "Brazilian Federal University" being composed of some faculties (such as "Faculty of Technology of UFES" and the "Faculty of Law of UFES") and a central management unit. The "Faculty of Technology of UFES" is composed of some departments such as the "Computer Science Department of UFES" and the "Electrical Engineering Department of UFES". Some of the types that composes the structure organizational model of UFES are represented in Fig. 6 with a white background.

B. Roles and Allocations

Fig. 7 presents the concepts related with the agents that compose the organization and the types of roles they may play (types shaded in light grey). We are concerned in this fragment with the roles that persons (instances of "Natural Person") play, first of all as a member of a formal organization, and then when they are given more specific places in organizational unit (organizational unit member).

To become a "Formal Organization Member", a person must be admitted by a formal organization, giving rise to an "Admission". Analogously, the association between a person (playing the role of "Unit Member") and an "Organizational Unit" is given by an "Assignment". Thus, both "Admission" and "Assignment" specialize the UFO notion of "Relator" being instances of "Relator Universal". "Employer" and "Allocation Unit" specialize, respectively, "Formal Organization" and "Organizational Unit", being both instances of "Role", while "Formal Organization Member" and "Unit Member" are instances of "Role" that specialize "Natural Person". Further, since "Natural Person" is a specialization of "Physical Agent" both, "Formal Organization Member" and "Unit Member" (indirectly) specialize "Physical Agent".

In order to play a particular role in an organizational unit, a person needs to be a formal organization member first. To capture this constraint, the assignments are tied with the admission that made the individual a member of the organization, which is defined through the relationship "refers to". An assignment *a* may only *refer to* an admission *b* iff: (i) the person that plays the role of "Unit Member" in *a* is the same person that plays the role of "Formal Organization Member" in *b*, and (ii) the "Allocation Unit" of *a* is part of the "Formal Organization" mediated by *b*. Further, since to play the role of "Unit Member" it is necessary to be a formal organization member, "Unit Member" is defined as a specialization of "Formal Organization Member".

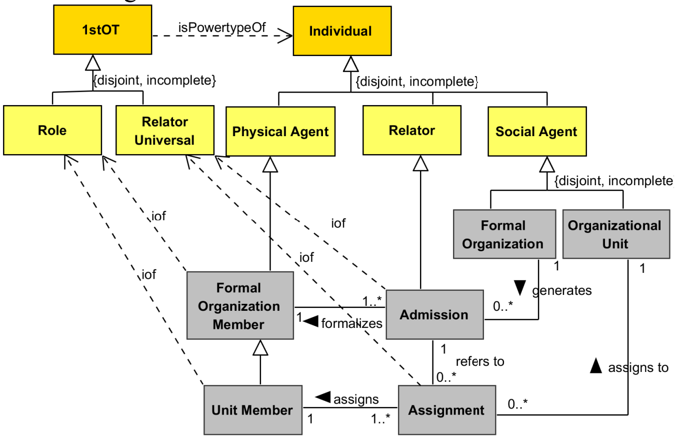

Fig. 7. Fragment of the core organizational structure ontology that copes with agents that compose the organization and types of roles they may play.

Different types of roles are relevant in the scope of different types of formal organizations. For example, in a university, employee types such as "Professor" and "Management Analyst" become relevant, while in a hospital employee types such as "Doctor" and "Nurse" may be defined. Therefore, our ontology includes the notion of "Formal Organization Member Type". Analogously, to fill the necessity of identifying different types of roles that are played in the context of different organizational units we propose the notion of "Business Role".

Both "Formal Organization Member Type" and "Business Role" are *second-order* types whose instances are roles. Thus, both specialize the UFO notion of "Role". Instances of "Employee Type" are specializations of "Formal Organization Member" such that each organization member is instance of at least one "Formal Organization Member Type". Thus, "Formal Organization Member Type" *completely characterizes* "Formal Organization Member". Similarly, since instances of "Business Role" specialize "Unit Member" such that each unit member is instance of at least one business role, we state that "Business Role" *completely characterizes* "Unit Member". Fig. 8 illustrates this. Types that are part of the core ontology are shaded in dark gray while the domain-specific types are in light gray.

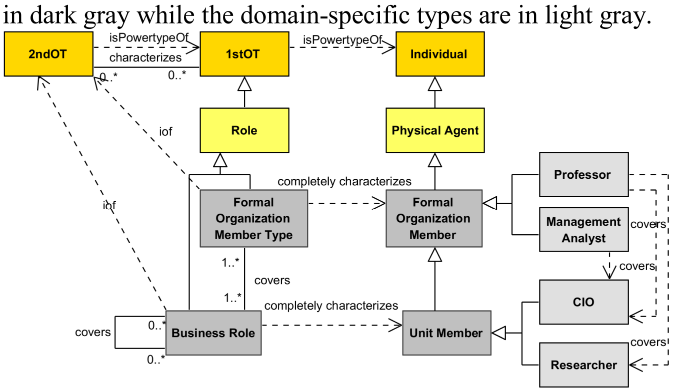

Fig. 8. Illustrating the relation between business roles and formal organization member types (the *cover* relation).

Specific employee types define the set of roles (business roles) that a typified employee can occupy in the organization. Business roles define more specific capabilities, duties and prerogatives possibly in the scope of organizational units. Members who are instances of an (instance of) "Employee Type" may play the roles that *are covered by* such employee type. For example, we may state that, in the context of a university, both professors and management analysts may play the role of department dean while only professors may play the role of researchers, i.e., both "Professor" and "Management Analyst" *cover* "Department Dean" but only "Professor" *covers* "Researcher".

Thus, we can define that if a role *x covers* a role *y* it means that: (i) instances of *x* are potential instances of *y* in the sense that individuals that plays the role *x* may also play the role *y*; and (ii) for every instance *i* of *y* we have that *i* is instance of *x* or there is another role *z* such that *z* covers *y* and *i* is instance of *z*. Note that it is also possible to define cover relations between business roles (see Fig. 8).

Each "Employee Type" is admitted in the context of at least one "Formal Organization Type" and for each "Formal Organization Type" there may be the necessity of certain employee Types. Thus, from the combination between a formal organization type and an employee type arises the concept of "Admission Type". The instances of "Admission Type" specializes "Admission" prescribing the employee types that are admissible in each formal organization type. Thus, "Admission Type" completely characterizes "Admission" and specializes the UFO notion of "Relator". The admission types allows the prescription of the possible allocations of an organization according to its type. For instance, we can define that a University must have, at least, twenty professors and two management analysts. Fig. 9 illustrates this scenario.

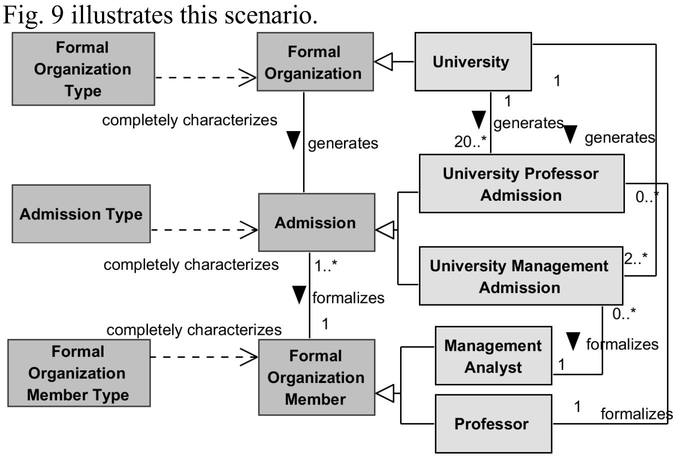

Fig. 9. Illustrating the notion of "Admission Type": "University Professor Admision" is an instance of "Admision Type" and specializes the concept of "Admission" considering the specific domain of universities (core ontology types are shaded in dark gray and the domain-specific ones are in light gray).

Analogously, each "Business Role" is played in the context of at least one "Unit Type" and for each "Unit Type" there may be the necessity of certain business roles. Thus, from the combination between a unit type and business role arises the concept of "Assignment Type". The instances of "Assignment Type" specialize "Assignment" prescribing the assignment types that are admissible in each unit type. Thus, "Assignment Type" completely characterizes "Assignment" and specializes the UFO notion of “Relator”. The assignment types allows the prescription of the possible allocations of a unit according to its type. For instance, we can define that each department must have one member playing the role of department dean. Fig. 10 illustrates this scenario. Recall that, to play a business role in a unit, (i) the person must be first admitted in the formal organization to which such unit belongs and (ii) the business role must be covered by the type instantiated by the person in the scope of its admission to the organization..

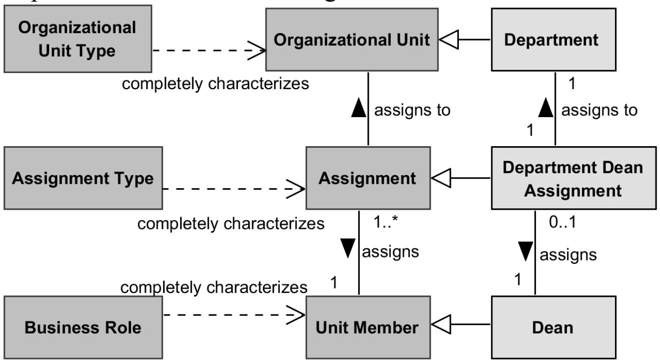

Fig. 10. Illustrating the notion of “Assigment Type”: “Department Dean Assignment” is an instance of “Assignment Type” and specializes the concept of “Assigment” considering the specific domain of universities (core ontology types are shaded in dark gray and the domain-specific ones are in light gray).

V. Related Work

As we have argued in this paper, a suitable conceptualization for organizational structure spans multiple levels of classification, requiring a multi-level modeling approach. In this section, we discuss the relation between MLT and other multi-level modeling approaches, focusing on the consequences of our choice of multi-level modeling approach, and contrasting it with alternative multi-level modeling approaches in the literature (including powertypes, materialization and deep instantiation). Later, we position our core ontology with respect to other organizational structure ontologies and modeling approaches.

A. Multi-Level Modelling Approaches

Two early attempts to address multi-level modeling, namely *power types* [18] [19] and *materialization* [25], raised from the identification of patterns to represent the relationship between a class of categories and a class of more concrete entities. The notion of *power types* was adopted in the object-oriented model community and *materialization* has been developed in the database community. Despite the different origins, they are based on similar conceptualizations [26] addressing similar concerns. Both approaches establish a relationship between two types such that the instances of one are specializations of another. The power type approach was incorporated in the UML [27], and the language currently includes a *power type association* that relates a classifier (power type) to a generalization set composed by the generalizations that occur between the base classifier and the instances of the power type. Because of its dependence on the generalization set construct, the UML power type pattern can only be applied when specializations of the base type are explicitly modeled (otherwise there would be no generalization set). We consider this undesirable, as it would rule out simple models that are possible in our approach, e.g., one defining “Unit Type” as a power type of “Unit”, without forcing the modeler to enumerate the instances of “Unit Type”. This feature of MLT is thus important here to enable us to express our core ontology independently of particular instances of “Unit Type”. Further, approaches based on the power type pattern do not allow the modeler to determine what kind of relation links higher order type and base type (*is powertype of, characterizes*). With MLT, we are able to capture the precise semantics of the relation between higher order type and base type, which is a more adequate choice for a semantic foundation considering that our focus is on representing a conceptualization as accurately as possible.

A more recent approach to deal with multi-level modeling is the deep instantiation approach of Atkinson and Kühne [10] [26]. Deep instantiation is proposed as a means to provide for multiple levels of classification whereby an element at some level can describe features of elements at each level beneath that level. The authors consider that the main benefit of deep instantiation is to support multi-level modeling without the need of introducing the required base type in the power type pattern, which they consider superfluous [26]. For example, using this approach it is possible to define unit types, such as “Department” and “Division”, omitting the notion of “Unit” from the domain model. To avoid representing the notion of “Unit”, all properties of it are defined in “Unit Type” as attributes with potency of 2. For instance, supposing that every unit has a phone number assigned to it, an attribute “phone number” is defined in “Unit Type” with potency of 2. When an instance of “Unit Type” is created (e.g. “Department”), the attribute “phone number” is added to it with potency of 1, meaning that their instances must have values assigned to it. We believe the adoption of this strategy trades the semantic clarity for reduction of model size, since it omits the base type and hides the inheritances between the omitted base type (“Unit”, in this example) and the instances of higher-order type (in this case, “Unit Type”). Further, as discussed in [12], conceptual models should always include kinds that define the principle of identity of individuals (in the example this type is “Unit”). If these types are omitted (and incorporated into higher-order types), the source of the principle of identity becomes hidden. Another important consequence of omitting a base type is that we become unable to express whether the instances of a higher-order type are disjoint types (i.e., we are unable to distinguish which form of *characterization* would apply). We are also prevented from determining metaproperties of the base type (such as e.g., rigidity). Using the MLT-UFO combination, we have been able to settle these metaproperties, by making every first-order type of our core ontology an instance of a second-order type in UFO’s taxonomy of universals. For example, this makes it clear that “Formal Organization” and “Organizational Unit” are rigid universals, and thus apply *necessarily* to the social agents that instantiate them. In contrast, “Formal Organization Member” and “Unit Member” are explicitly identified as anti-rigid universals applying *contingently* to the agents that instantiate them.

The *multi-level objects* (or *m-objects)* [9] is another multi-level modeling approach that applies the notion of *deep instantiation*. Similarly to Atkinson and Kühne’s proposal the approach leads to a model with fewer elements, but prevents us from expressing important aspects of the first-order types.

B. Organizational Structure Ontologies

As discussed in [15], the organizational structure domain has been the subject of a number of ontologies since the end of the 90s, including initiatives such as The AIAI Enterprise Ontology (EO) [6], the organization ontology for the TOVE enterprise model[7], the W3C Org Ontology [8], among others.

The AIAI Enterprise Ontology (EO) [6] is described in natural language and is based on formalized meta-ontology, with good coverage of concepts related to organization structure [15]. As discussed in [15], EO includes a direct relationship between a "person" and an "organisation unit" ("working for"), without the intermediary of roles or positions they play in the scope of an "organizational unit". In case a person plays multiple roles it is not possible to define which role is played in the context of each "organisation unit". Further, EO does not provide second-order notions such as "Organization Type", "Unit Type" or "Roles". Instead, it defines fixed sets of types and roles for specific domains (e.g., "Vendor", "Customer", "Reseller"). This makes it less general than the core ontology discussed here.

A similar remark can be made with respect to the organization ontology for the TOVE enterprise model [7] which chooses for a fixed hierarchical structure for the organizational with three levels: "organization", "division" and "sub-division". Choosing fixed roles and types restricts the applicability of these ontologies, making them unsuitable to any organizational contexts not structured according to these three levels. We take a different approach and aim here at a more general ontology while employing a hierarchical approach to cater for domain-specificity. Specific structures with instances of higher-order types can appear at a lower level of specificity, e.g., in an enterprise-specific ontology that extends the core organizational ontology for a particular organizational setting.

The W3C Org Ontology [8] does not fix specific roles and types, but it also does not provide second-order concepts (which would be comparable to our "Formal Organization Type" and "Unit Type"). It provides thus no facility for its users to create first-order types and their conceptual distinctions.

The concepts of the core organization structure ontology we propose here are based on the notions defined in the O3 ontology [23] [15]. Our ontology differs from O3 in its focus: we are concerned here solely with the most general concepts for organizational structure highlighting the aspects that are specific to multi-level modeling. For instance, we introduce the second-order concept of "Organizational Unit Type" and avoid fixing the distinctions used to specialize the notion of "Organizational Unit" such as the O3 distinctions between "structural" vs. "missionary units" and between "staff" vs. "line units". Following our hierarchical modeling approach, these distinctions can be accounted for in a lower-level ontology. For example, applying the basic pattern of MLT the O3 notion of "Structural Unit" can be defined as an instance of "Organizational Unit Type" and, simultaneously, as a specialization of "Organizational Unit". A topic for further investigation concerns the specification of other parts of O3 as an extension of this core ontology.

E-OPL [28] is another example of ontology for organizations founded in UFO. It includes some notions for organizational structure, and aims to provide a basis for an enterprise pattern language whose fragments can be selected flexibly. Both O3 and E-OPL are defined using a UML profile that incorporates the foundational distinctions of UFO using UML stereotypes (called OntoUML [12]). The only multi-level modeling support provided to OntoUML users is UML's support for *power types*. As a consequence, the limitations of the power type solution (previously discussed in section V.A.) also apply to O3 and E-OPL.

C. Organizational Structure Modelling Approaches

Many prominent enterprise architecture modeling approaches cater for the representation of organizational structures. For example, the Architecture of Integrated Information Systems (ARIS) [2] is an enterprise architecture framework that provides both a method for analysis and design of organizational aspects (including organizational structure) and a language for its representation. ARIS has a significant number of modeling constructs for organizational structure, including second-order notions such as organization unit type and unit member type. Nevertheless, as pointed out in [1], ARIS lacks a sound semantic foundation and many of its modeling language constructs present problems such as redundancies and semantic overload [12]. As discussed in [1] these problems also apply to the second-order notions. The ARIS metamodel seems to have evolved in an *ad hoc* manner and some of the second-order notions seem to have been introduced irregularly. For example, there is no support to represent the specialization of organizational unit types (while there is support to represent the specialization of "Person Types"); the notion of "Person Type" may be applied indiscriminately to persons and organization units alike; "positions" can be instances of "position types" but also "organizational unit types". Please refer to [1] for an in depth discussion on the semantics of ARIS organizational structure elements.

Another widely employed EA modeling language that includes organizational structure elements is ArchiMate [3]. A strength of the language is the broad coverage of a wide number of aspects of EA, and the possibility to describe relations between the various aspects. Nevertheless, the emphasis on providing an overview of relations seems to have led to a less sophisticated treatment of some aspects, and that includes the active structure domain [15]. In [15] the authors conducted a semantic analysis of the fragment of the ArchiMate metamodel related with the representation of active structure revealing some problems caused by its lack of a sound semantics foundation. Moreover, such fragment of ArchiMate does not provide support to the specification of second-order notions such as organizations types or business roles types.

Similarly to our approach, the Department of Defense Architecture Framework (DoDAF) [4] provides an account for the second-order notions of organizations types and role types (in its Operational Viewpoint OV-4) with a multi-level approach based on the IDEAS Foundational Ontology. Differently from our approach, they have chosen to consider the membership relations between organizations and their employees as "whole part" relations, which are specializations of the formal notion of "tuple" in IDEAS. A tuple is defined as an ordered pair of two things. In contrast, our approach considers the membership relations as specialization of the UFO notion of “relator”. In this view, the relationship between members and organizations (as well as the relationship between unit members and units) can: qualitatively change while maintaining their identity; be the subject of modal properties; be characterized by having both essential and accidental properties [29]. None of this is possible in the case in which relationships are reduced to tuples (see [29] for a full discussion on the benefits of this foundational account to relationships).

VI. Final Considerations

Aiming to address the lack of a suitable foundation for organizational structure modeling, this paper presents a core organizational structure ontology built with the combination of UFO and MLT. The foundational distinctions provided by UFO have allowed us to address some ontological issues concerning the core ontology concepts (e.g., anti-rigidity of role types, reification of relators) while the use of MLT has allowed us to address higher-order types and provide us with a sound basis to establish the relation between foundational ontology, core ontology and their specializations (e.g., an enterprise-specific ontology).

The result is a hierarchical modeling approach. The core organizational structure ontology may be extended by enterprise-specific ontologies that simultaneously instantiate and specialize the core ontology distinctions with the concepts that are required in a particular organizational setting. Such a hierarchical approach is required to cope with the large diversity in organizational structures and structuring approaches.

The core ontology has been defined as a simplified version of O3 focusing on the general concepts and highlighting the aspects that are specific to multi-level modeling. It adds to O3 *second-order types* to map the notions of “Unit Type”, “Organization Type” and “Assignment Type”. It addresses second-order types that are not covered by ArchiMate, the W3C Org Ontology, the TOVE Enterprise Ontology and the AIAI Enterprise Ontology. These approaches propose a fixed set of organization and role types to their users, and thus cannot accommodate variations of enterprise settings.

Topics for further investigation concern the specification of other parts of O3 as an extension of this core ontology, and the specification of domain- or industry-specific models as an extension of O3 applying thus the proposed hierarchical modeling approach. Other topics of further investigation include revisiting the ontological analysis of ARIS conducted in [1] and the extension of ArchiMate proposed in [23] in light of UFO-MLT and the core ontology presented here.

VII. Acknowledgements

This research is funded by the Brazilian Research Funding Agencies CNPq (grants number 311313/2014-0, 485368/2013-7 and 461777/2014-2) and CAPES/CNPq (402991/2012-5). Victorio A. Carvalho is funded by CAPES.

References

[1] J. Santos, J. P. A. Almeida, and G. Guizzardi, “An ontology-based analysis and semantics for organizational structure modeling in the ARIS method,” Inf. Syst., vol. 38, no. 5, pp. 690–708, 2013.

[2] A. W. Scheer, “Business process modeling,” 3rd ed, Springer,1999.

[3] The Open Group, “ArchiMate Version 2.1 Technical Standard,” 2012. Available at http://pubs.opengroup.org/architecture/archimate2-doc/

[4] US Department Of Defense, “DoDAF Specification,”. Avaiable at http://dodcio.defense.gov/Portals/0/Documents/DODAF/DoDAF_v2-02_web.pdf

[5] ISO/ITU-T, “Open Distributed Processing - Reference Model” [s.l: s.n.]

[6] M. Uschold, M. King, S. Moralee, and Y. Zorgios, “Enterprise Ontology specification,” 1998. Available at http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html.

[7] M. S. Fox, M. Barbuceanu, M. Gruninger, and J. Lin, “Simulating Organizations,” MIT Press, pp. 131–152, 1998.

[8] W3C, “Org Ontology specification,” 2014. Available at http://www.w3.org/TR/2014/REC-vocab-org-20140116/.

[9] B. Neumayr, K. Grün, and M. Schrefl, “Multi-level domain modeling with m-objects and m-relationships,” in 6th Asia-Pacific Conf. on Conceptual Modeling, 2009.

[10] C. Atkinson, and T. Kühne, “The Essence of Multilevel Modeling,” in 4th International Conf. on the Unified Modeling Language, 2001.

[11] J. P. A. Almeida, G. Guizzardi, and J. Santos, “Applying and Extending a Semantic Foundation for Role-Related Concepts in Enterprise Modeling,” Enterprise Information Systems, vol.3, pp. 253-277, 2009.

[12] G. Guizzardi, “Ontological Foundations for Structural Conceptual Models,” University of Twente, 2005.

[13] J. C. Nardi, et al., “A commitment-based reference ontology for services,” Information Systems (Oxford), vol. 51, p. 1, 2015.

[14] J. C. Nardi, R. A. Falbo, and J. P. A. Almeida,“An Ontological Analysis of Service Modeling at ArchiMate Business Layer,” in Proc. 18th Intl. Enterp. Distr. Obj. Computing Conf. (EDOC 2014) , pp. 92-100, 2014.

[15] D. Pereira and J. P. A. Almeida, “Representing Organizational Structures in an Enterprise Architecture Language,” in 6th Workshop on Formal Ontologies meet Industry (FOMI 2014), pp. 1-12, 2014.

[16] V. A. Carvalho, and J. P. A. Almeida, “Towards a Well-Founded Theory for Multi-Level Conceptual Modeling,” available at http://nemo.inf.ufes.br/mlt, 2015, unpublished.

[17] C. Atkinson, and T. Kühne, “Meta-level Independent Modeling,” in. Intl Workshop “Model Engineering” (in conj. with ECOOP’2000), 2000.

[18] L. Cardelli, “Structural Subtyping and the Notion of Power Type,” in 15th ACM Symp. of Principles of Programming Languages, pp. 70-79, 1988.

[19] J. Odell, “Power types,” in Journal of Object-Oriented Programing, v. 7(2), pp. 8-12, 1994.

[20] J. P. A. Almeida, and G. Guizzardi,“An Ontological Analysis of the Notion of Community in the RM-ODP Enterprise Language,” Computer Standards & Interfaces, vol. 34, p. 1, 2012.

[21] V. A. Carvalho, J. P. A. Almeida, C. M. Fonseca and G. Guizzardi, “Extending the Foundations of Ontology-based Conceptual Modeling with a Multi-Level Theory,” in 34th Intl. Conf. on Conceptual Modeling (ER 2015), 2015.

[22] G. Guizzardi, R. S. S. Guizzardi, and R. A. Falbo, “Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology,” in XI Iberoamerican Workshop on Requirements Engineering and Software Environments (IDEAS 2008), pp. 127-140, 2008.

[23] D. C. Pereira, “Representing Organizational Structures in Enterprise Architecture: An Ontology-Based Approach,” Federal University of Espírito Santo, 2015

[24] A. Etzioni, “Modern organizations,” Prentice-Hall, 1964.

[25] A. Pirotte, E. Zimanyi, D. Massart, and T. Yakusheva, “Materialization: a powerful and ubiquitous abstraction pattern,” in 20th Int. Conf. Very Large DataBases, pp. 630–641, 1994.

[26] C. Atkinson, and T. Kühne, “Reducing accidental complexity in domain models,” Software & Systems Modeling, v. 7(3), pp 345-359, 2008.

[27] OMG, “UML Superstructure Specification – Version 2.4.1”, 2011.

[28] R. A. Falbo, F. B. Ruy, G. Guizzardi, M. P. Barcellos, and J. P. A. Almeida, “Towards an enterprise ontology pattern language,” in 29th Annual ACM Symp. on Applied Computing, pp. 323-330, 2014.

[29] N. Guarino, and G. Guizzardi, “We need to Discuss the Relationship: Revisiting Relationships as Modeling Constructs,” in 27th Intl. Conf. on Advanced Information Systems Engineering, 2015