Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology

Giancarlo Guizzardi, Ricardo Falbo, Renata S.S.Guizzardi

Computer Science Department
Federal University of Espírito Santo (UFES), Vitória, Brazil
{gguizzardi,falbo,rguizzardi}@inf.ufes.br

Abstract. *Foundational Ontologies* are theoretically well-founded domain-independent systems of categories that have been successfully used to improve the quality of conceptual modeling languages and models. In this paper, we present the latest developments in the UFO ontology. Moreover, we elaborate on the relevance of these foundational ontologies in the development of domain ontologies by showing a case study in the software process domain.

Keywords: Foundational Ontology, Ontology in Software Engineering, Software Process

1 Introduction

Ontologies have been acknowledged as a useful conceptual tool in computer science since the late sixties, chiefly in the areas of data modeling and artificial intelligence [1]. In the past seven years, an explosion of works related to ontology has happened in several different fields of computer science, chiefly motivated by the growing interest on the Semantic Web, and by the key role played by them in that initiative. In the area of Software Engineering, domain ontologies have been used mainly as an enhanced representation for what is termed a domain model in the field of Domain Engineering. Ontologies for domains such as software process, software quality, software resource management, software project risks, among many others have been used for: (i) deriving reusable domain-specific frameworks [2]; (ii) integrating knowledge in semantic software environments [3].

An important point that should be emphasized is the difference in the senses of the term ontology when used by the data modeling community, on one side, and artificial intelligence, software engineering and semantic web communities on the other [1]. In data modeling and related areas (e.g., organizational engineering), the term has been used in ways that conform to its definitions in philosophy, namely, as a philosophically well-founded domain-independent system of formal categories that can be used to articulate domain-specific models of reality. In contrast, in the other areas aforementioned, the term ontology is, in general, used as: (i) a concrete engineering artifact designed for a specific purpose and without paying much attention for foundational issues; (ii) a representation of a singular domain (e.g., molecular biology, finance, logistics, ceramic materials) expressed in knowledge representation languages (e.g., RDF, OWL, F-Logic) or conceptual modeling grammars (e.g., UML, EER).

Ontologies, in the philosophical sense, have been developed in philosophy since Aristotle's theory of Substance and Accidents and, recently, a number of such theories have been proposed in the area of Applied Ontology in Computer Science under the name *Foundational Ontologies*. In this paper, we discuss a particular Foundational Ontology named *UFO (Unified Foundational Ontology)*.

UFO has been developed based on a number of theories from Formal Ontology, Philosophical Logics, Philosophy of Language, Linguistics and Cognitive Psychology. The core of this ontology (UFO-A) is presented in depth and formally characterized in [1]. Moreover, in a number of publications, UFO has been successfully employed to evaluate, re-design and integrate the models of conceptual modeling languages as well as to provide real-world semantics for their modeling constructs (e.g., [1,4-6]). In [1], a complete evaluation and re-design of the UML 2.0 metamodel using UFO is presented. In [5], we show how a modeling language based on this ontology can be used to address a number of semantic interoperability problems which cannot be handled by semantic web languages such as OWL and RDF. Finally, in [6], we evaluate, redesign and integrate two well-known modeling languages which are used in different phases of agent-oriented development processes, namely, TROPOS (for requirements engineering and early design) and AORML (for detailed design).

The purpose of this article is two fold. Firstly, we present a new version of two fragments of this foundational ontology, named UFO-B and UFO-C. UFO-B is an ontology of events; UFO-C builds on top of A and B to systematized social concepts such as plan, action, goal, agent, intentionality, commitment, appointment, among many others. Secondly, we elaborate on the relation between the two senses of ontologies aforementioned by illustrating the importance of foundational ontologies in the development of domain ontologies. In particular, we demonstrate how UFO has been used to evaluate, re-design and to give real-world semantics to an ontology in the software engineering domain, namely, the software process ontology which is the core of the ODE Ontology-Centered Software Development Environment [3].

The remaining of this article is organized as follows: in section 2, we present a small fragment of UFO-A discussing only those categories which are essential for the understanding of the sections to follow. In section 3 and 4 we introduce the new developed theories for UFO-B and C, respectively. Section 5 discusses the original ODE Software Process Ontology and demonstrates the evaluation and re-design of this ontology when mapped to (interpreted in terms of) UFO. Finally, section 6 elaborates on some final considerations of this article.

2 UFO-A: An Ontology of Endurants

A fundamental distinction in this ontology is between the categories of ***Particular (Individual)*** and ***Universal (Type)***. Particulars are entities that exist in reality possessing a unique identity. Universals, conversely, are pattern of features, which can be realized in a number of different particulars. ***Substances*** are existentially independent particulars. Examples include ordinary mesoscopic objects such as an individual person, a dog, a house, a hammer, a car, Alan Turing and The Rolling Stones. The word ***Moment,*** in contrast, denotes, what is sometimes named a trope, or an individualized (objectified) property or property in particular. Therefore, in the scope of this work, the word bears no relation to the notion of time instant in colloquial language. A moment is an individual that can only exist in other individuals. Typical examples of moments are a color, a connection, an electric charge, a symptom, a covalent bond. Moments have in common that they are all dependent of other individuals (their bearers). An important feature that characterizes all *moments* is that they can only exist in other individuals (in the way in which, for example, electrical charge can exist only in some conductor, or that a covalent bond can only exist if those connecting atoms exist). To put it more technically, we say that moments are *existentially dependent* on other individuals. Existential dependence can also be used to differentiate intrinsic and relational moments: ***intrinsic moments*** are dependent of one single individual (e.g., color, a headache, a temperature); ***relators*** depend on a plurality of individuals (e.g., an employment, a medical treatment, a marriage). Finally, we consider here the categories of ***substantial universal*** and ***moment universal***. Examples of the former include Apple, Planet and Person. Examples of the latter include Color, Electric Charge and Headache.

An attempt to model the relation between intrinsic moments and their representation in human cognitive structures is presented in the theory of *conceptual spaces* introduced in [7]. The theory is based on the notion of ***quality structure***. The idea is that for several perceivable or conceivable moment universals there is an associated quality structure in human cognition. For example, height and mass are associated with one-dimensional structures with a zero point isomorphic to the half-line of nonnegative numbers. Other properties such as color and taste are represented by multi-dimensional structures. In [7], the perception or conception of an intrinsic moment can be represented as a point in a quality structure. Following [8], this point is named here a ***quale***. Quality structures and qualia are together with sets, number and propositions examples of ***Abstract Particulars***.

Relations are entities that glue together other entities. In the philosophical literature, two broad categories of relations are typically considered, namely, *material* and *formal* relations [9]. Formal relations hold between two or more entities directly, without any further intervening individual. In principle, the category of formal relations includes those relations that form the mathematical superstructure of our framework including *existential dependence*, *part-of* (<), *subset-of*, *instantiation*, among many others not discussed here [1]. *Material relations*, conversely, have material structure of their own and include examples such as *working at*, *being enrolled at*, and *being connected to*. Whilst a formal relation such as the one between Paul and his knowledge x of Greek holds directly and as soon as Paul and x exist, for a material relation of *being treated in* between Paul and the medical unit MU_1 to exist, another entity must exist which *mediates* Paul and MU_1. We name these entities *relators*. Relators are individuals with the power of connecting entities. For example, a medical treatment connects a patient with a medical unit; an enrollment connects a student with an educational institution; a covalent bond connects two atoms. The notion of relator (relational moment) is supported by several works in the philosophical literature [1,9] and, the position advocated here is that they play an important role in answering questions of the sort: what does it mean to say that John is married to Mary? Why is it true to say that Bill works for Company X but not for Company Y?

Suppose that John *is married to* Mary. In this case, we can assume that there is an individual relator (relational moment) m_1 of type *marriage* that mediates John and Mary. In this case, there are many moments that John acquires by virtue of being married to Mary. For example, imagine all the legal responsibilities that John has in the context of this relation. These newly acquired properties are intrinsic moments of John which, therefore, are existentially dependent on him. However, these moments also depend on the existence of Mary. We name this type of moment ***externally dependent moment***, i.e., externally dependent moments are intrinsic moments that inhere in a single individual but are existentially dependent on (possibly multiple) other individuals. The relator marriage in this case is the sum of all externally dependent moments that John and Mary acquire by virtue of being married to each other.

Finally, we consider here the notion of ***Situations*** proposed, for example, in [9]. Situations are special types of endurants. These are complex entities that are constituted by possibly many endurants (including other situations). Situations are taken here to be synonymous to what is named *state of affairs* in the literature, i.e., a portion of reality that can be comprehended as a whole. Examples of situations include *"John being with fever and influenza", "John being in the same location as Paul while Mary is in the same location as David", "Mary being married to Paul who works for the University of Twente"*. Finally, we define a relation of *"being present at"* between endurants and the situations they constitue. For instance, we can state that both the substantial John, and the intrinsic moments m_1 (John's Fever) and m_2 (John's influenza) are present in situation s_1:*"Jonh being with fever and influenza"*. For a more detailed discussion on our view of situations and contexts, one should refer to [10].

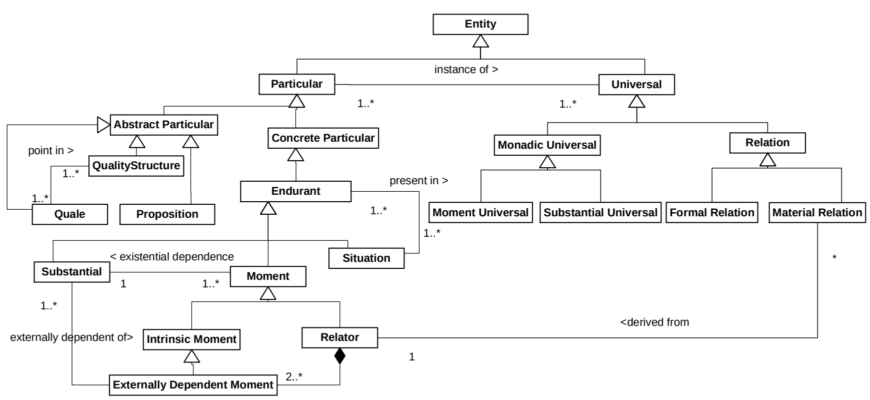

Fig.1. A Fragment of a Foundational Ontology of Endurants (UFO-A).

3 UFO-B: An Ontology of Perdurants

UFO-B makes a distinction between enduring and perduring individuals (henceforth named *endurants* and *perdurants*). Classically, this distinction can be understood in terms of their behavior w.r.t. time. Endurants are said to be wholly present whenever they are present, i.e., they *are in time*, in the sense that if we say that in circumstance c_1 an endurant *e* has a property P_1 and in circumstance c_2 the property P_2 (possibly incompatible with P_1), it is the very same endurant *e* that we refer to in each of these situations. Examples of endurants are a house, a person, the moon, a hole, an amount of sand. For instance, we can say that an individual John weights 80kg at c_1 but 68kg at c_2. Nonetheless, we are in these two cases referring to the same particular John.

Perdurants are individuals composed of temporal parts, they *happen in time* in the sense that they extend in time accumulating temporal parts. Examples of perdurants are a conversation, a football game, a symphony execution, a birthday party, the Second World War and a business process. Whenever a perdurant is present, it is not the case that all its temporal parts are present. For instance, if we consider a business process "buy product" at different time instants when it is present, at each of these time instants only some of its temporal proper parts are present. As a consequence, perdurants cannot exhibit change in time in a genuine sense since none of its temporal parts retain their identity through time.

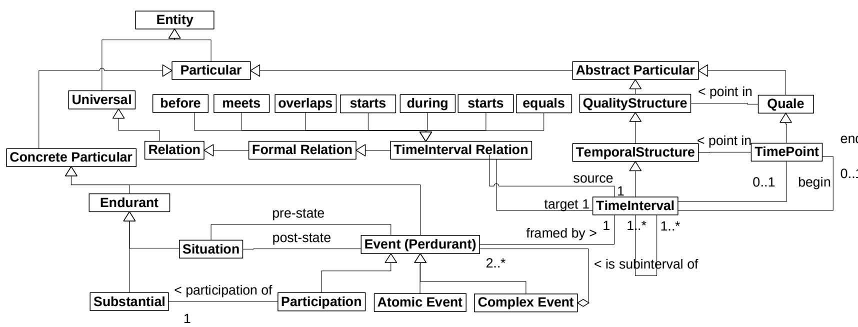

Fig.2. A Fragment of a Foundational Ontology of Perdurants (UFO-B).

Figure 2 depicts a fragment of our ontology of perdurants named UFO-B. The main category on this ontology is ***Event*** (Perdurant, Occurrent). Events can be Atomic or Complex, depending on their mereological structure, i.e., whilst atomic events have no improper parts, complex events are aggregations of at least two events (that can themselves be atomic or complex). Events are possible transformations from a portion of reality to another, i.e., they may change reality by changing the state of affairs from one (pre-state) situation to a (post-state) situation. Events are ontologically dependent entities in the sense that they existentially depend on their participants in order to exist. Take for instance the event *e*: *the stabbing of Caesar by Brutus*. In this event we have the participation of Caesar himself, of Brutus and of the knife. In this case, *e* is composed of the individual participation of each of these entities. Each of these participations is itself an event that can be complex or atomic but which existentially depends on a single substantial. It is important to emphasize that being atomic and being instantaneous are orthogonal notions in this framework, i.e., atomic participations can be time-extended as well as an instantaneous event can be composed of multiple (instantaneous) participations. In summary, the model of figure 2 depicts these two aspects on which events can be analyzed, namely, as time extended entities with certain (simple or complex) mereological structures, and as ontologically dependent entities which can comprise of a number of individual participations.

As in [8], we have that all spatial properties of events are defined in terms of the spatial properties of their participants. In contrast, all temporal properties of substantials are defined in terms of the events they participate. Analogous to what has been discussed for endurants, the temporal properties of events have their values taken (their qualia) by projecting these properties into a quality structure. We here take the time conceptual space to be a structure "composed of" Time Intervals. Time intervals themselves are "composed of" Time Points. Time points could be represented as real numbers and Time Intervals as sets of real numbers. However, they could also be interpreted as sui generis entities such as Chronoids and Time Boundaries in GFO [9]. In other words, we avoid making unnecessary ontological commitments at this point. Additionally, we admit: (i) intervals that are delimited by begin and end points as well as open intervals; (ii) continuous and non-continuous intervals; (iii) intervals with and without duration (instants). In particular, this model allows a diversity of temporal structures such as linear, branching, parallel and circular time. For the case of ordered structures we have considered the so-called *Allen Relations* [11] between intervals from which corresponding relations between events can be derived. It is important to emphasize that it is outside the focus of this article to further elaborate on the nature of these temporal structures. For this reason, we aim at characterizing only the properties which are germane to the objectives pursued here.

4 UFO-C: An Ontology of Social Entities

The third layer of the Unified Foundational Ontology is an ontology of social entities (both endurants and perdurants) built on top of UFO-A and UFO-B. A fragment of this ontology is shown in Figure 3. We start by making a distinction between Agentive and Non-agentive substantial particulars, termed here ***Agents*** and ***Objects***, respectively. Agents can be physical (e.g., a person) or social (e.g., an organization, a society). Objects can also be further categorized in physical and social objects. Physical objects include a book, a tree, a car; Social objects include money, language and ***Normative Descriptions***. A normative description defines one or more rules/norms recognized by at least one social agent and that can define nominal universals such as social moment universals (e.g., social commitment types), social objects (the crown of the king of Spain) and social roles such as president, prime minister, PhD candidate or pedestrian. Examples of normative descriptions include the Italian Constitution, the University of Twente PhD program regulations, but also a set of directives on how to perform some actions within an organization (a description of a plan [12]).

Agents are substantials that can bear special kinds of moments named ***Intentional Moments***. As argued in [13], intentionality should be understood in a much broader context than the notion of "intending something", but as the capacity of some properties of certain individuals to refer to possible situations of reality. Every intentional moment has a type (e.g., ***Belief***, ***Desire***, ***Intention***) and a propositional content. The latter being an abstract representation of a class of situations referred by that intentional moment. Thus, "intending something" is a specific type of intentionality termed

Intention. The propositional content of an Intention is a ***Goal***. The precise relation between an intentional moment and a situation is the following: situation in reality can satisfy the propositional content of an intentional moment (i.e., satisfy - in the logical sense – the proposition representing that propositional content). Beliefs can be justified by situations in reality. Examples include my belief that Rome is the Capital of Italy, and the Belief that the Moon orbits the Earth; Desires and Intentions can be fulfilled or frustrated. Whilst a desire expresses a will of an agent towards a state of affairs in reality (e.g., a Desire that Brazil wins the Next World Cup), intentions are desired state of affairs for which the agent commits at pursuing (internal commitment) (e.g., the Intention of going to a beach resort for the next summer break) [13,14]. For this reason, intentions cause the agent to perform **Actions**.

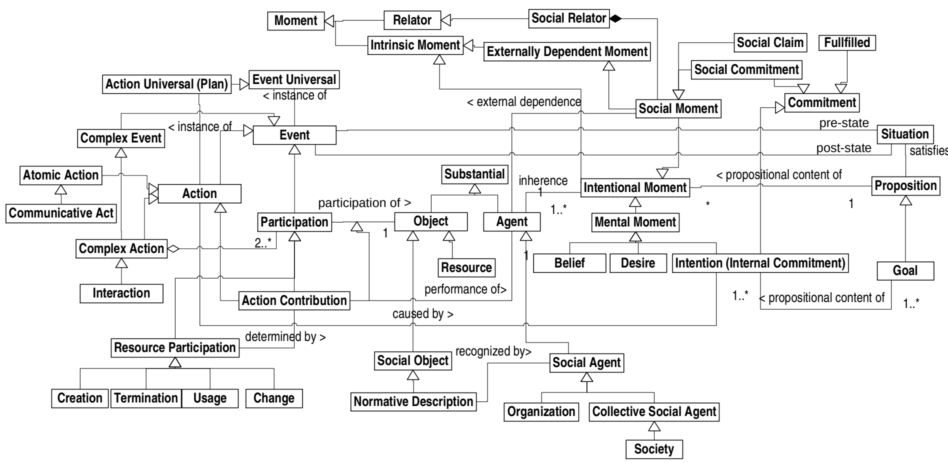

Fig.3. A Fragment of a Foundational Ontology of Social Entities (UFO-C).

Actions are intentional events, i.e., events which instantiate a ***Plan*** (Action Universal) with the specific purpose of satisfying (the propositional content of) some intention. Examples of actions include writing this paper, a business process, a communicative act. In particular, a ***Communicative Act*** (a speech act such inform, ask or promise) [13] is an example of an atomic action. As events, actions can be atomic or complex. A complex action is composed of two or more participations. These participations can themselves be intentional (i.e., be themselves actions) or unintentional events. For example, the stabbing of Caesar by Brutus includes the intentional participation of Brutus and the unintentional participation of the knife. In other words, following philosophical action theories [15], we take that it is not the case that any participation of an agent is considered an action, but only those intentional participations – termed here ***Action Contributions***. Only agents (entities capable of bearing intentional moments) can perform actions. An object participating in an action is termed a ***Resource***. A complex action composed of action contributions of different agents is termed an ***Interaction***. Two artists collaborating to create a sculpture is an example of an interaction, and so is a dialogue between two agents. In the former case, the sculpture as well as the tools and raw materials used to create it are examples of resources. Objects can participate in actions in different ways. In this article, we countenance four different modes of ***Resource Participation***, namely, ***Creation***, ***Termination***, ***Change*** or ***Usage***, which can be characterized as follows: Let *r* be a resource, *a* an action, and s_1, s_2 two situations such that they are the pre and post state of action *a*. Then we have that (i) <u>creation</u>**:** a resource participation of *r* in *ac* is a creation iff (*r is not present in* s_1) and (*r is present in* s_2) and (there is at least one action contribution *ac* that is also part of *a* and such that s_2 satisfies the propositional content of *ac*); (ii) <u>termination:</u> a resource participation of *r* in *a* is a termination iff (*r is present in* s_1) and (*r is not present in* s_2) and (there is at least one action contribution *ac* that is also part of *a* and such that s_2 satisfies the propositional content of *ac*); (iii) <u>change:</u> a resource participation of *r* in *a* is a change iff there is at least a moment *m* such that ((*m inheres in r in* s_1 and *m does not inhere in r in* s_2) OR (*m does not inhere in r in* s_1 and *m inheres in r in* s_2)) and (there is at least one action contribution *ac* that is also part of *a* and such that s_2 satisfies the propositional content of *ac*); (iv) a resource participation which is not any of the three aforementioned modes of participation is a usage participation.

As discussed in [6], a resource participation can be the cause of a *resource dependence* and the result of a *resource acquisition* between agents. In a resource acquisition from agent *B* to agent *A*, *A* gives permission of resource *r* to *B*. For this to happen, *A* must have the right to grant permission to agent *B* and, moreover, the right to grant the right mode of permission (e.g., to use, to modify). In other words, different modes of resource participations can be connected to different deontic consequences of the relation between agents. In summary, a resource participation in an action is the intended use, modification, termination or creation of an object in that action. Thus, for instance, we consider a resource change of resource *r* in action *a* when this change is the content of the intention of at least one of the agents in that action.

Communicative Acts can be used to create ***Social Moments***. In this view, language not only represents reality but also creates a part of reality [13]. Thus, social moments are types of intentional moments that are created by the exchange of communicative acts and the consequences of these exchanges (e.g., goal adoption, delegation [6]). For instance, suppose that I rent a car at a car rental service. When signing a business agreement, I perform a communicative act (a promise). This act creates a ***Social Commitment*** towards that organization: a commitment to *return the car in a certain state, etc* (the propositional content). Moreover, it also creates a ***Social Claim*** of that organization towards me w.r.t. that particular propositional content. Commitments/Claims always form a pair that refers to a unique propositional content. A ***Social Relator*** **(**or ***Social Bond*****)** is an example of a relator composed of two or more pairs of associated commitments/claims (social moments). Finally, a commitment (internal or social) is fulfilled by an agent *A* if this agent performs an action *x* such that the post-state of that action is a situation that satisfies that commitment.

4.1 The Distinction between Action Universals, Action Occurences and Scheduled Actions

In this section, we elaborate on some of the concepts discussed above with the intention of clarifying a distinction which is commonly blurred in several process models, namely, the distinction between Action Universals, Action Occurences and Scheduled

Actions. In figure 4 below, we present additional UFO-C ontological categories needed to discuss these distinctions.

We start by elaborating on the notions of commitments. Commitments (internal or social) can be ***Fulfilled*** or ***Unfulfilled***. Unfulfilled commitments can be ***Pending***, ***Dismissed*** or ***Broken***. A Social Commitment *C* is a commitment of an agent *A* towards another agent *B*, thus, as an externally dependent moment, we state that *C* inheres in *A* and is externally dependent on *B*. Differently from an internal commitment, in this case, *C* can only be dismissed by *B*. Internal commitments cause the agent to perform actions. Thus, following [14], we have that social commitments necessarily cause the creation of internal commitments, i.e., if I promise to bring you a specific book by tomorrow, asides from the commitment with you, I must also create the intention (internal commitment) of bringing the book by tomorrow.

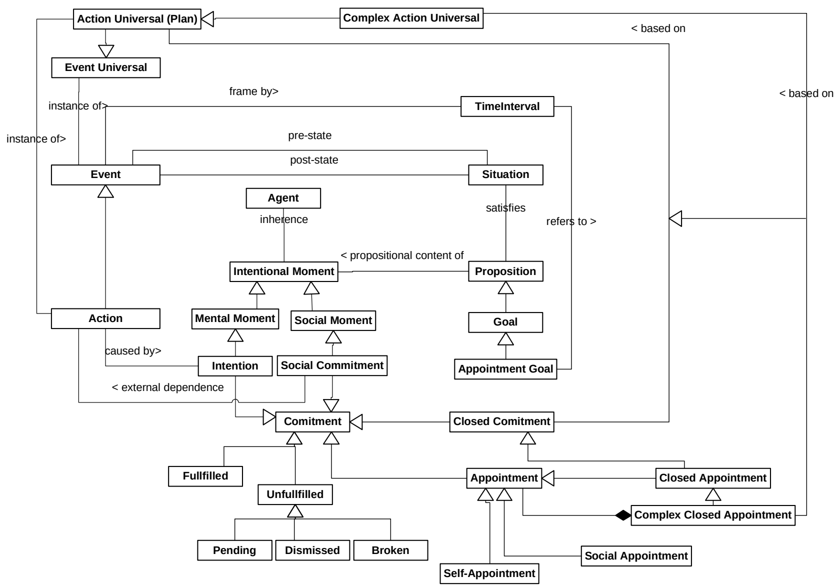

Fig.4. A Fragment of UFO-C dealing with Appointments.

As discussed in the previous section, a commitment (internal or social) is fulfilled by an agent *A* if this agent performs an action *a* such that the post-state of that action is a situation that satisfies the propositional content of that commitment. For many situations, there are a number of possible actions that can bring about that situation. We differentiate between ***Open*** and ***Closed commitments***, the difference being that, in the latter case, the agent must fulfil the commitment (i.e., bring about the desired situation) by executing a specific action. We state then that a closed commitment C is based on an Action Universal (Plan) *P* such that *C* is fulfilled by agent *A* iff *A* brings about a situation that satisfies the propositional content of *C* by executing an action *a* which is an instance of *P*. Open and Closed commitments can explain the notions of Open and Closed Delegation [14], respectively. These, in turn, can explain the difference between what is named *Goal* and *Plan Dependence* in requirements engineering languages such as TROPOS and i* [6].

A special type of Commitment is an ***Appointment***. An appointment is a commitment whose propositional content explicitly refers to a time interval. For instance, while “I will return the book to you” is a social commitment, “I will return the book between now and the end of this working week” is an appointment. An appointment can be either Internal (***Self-Appointment***) or a ***Social Appointment***. A ***Closed Appointment*** is a Closed Commitment whose propositional content explicitly refers to a time interval. A ***Complex Closed Appointment*** is based on a ***Complex Action Universal***. On one hand, a complex closed appointment *C* is composed of a number of commitments that should be achieved by executing a number of actions that are part of a complex action (instance of the complex action universal on which *C* is based). On the other hand, *C* stands for a number of commitments to enact a complex action universal by executing its specific sub-actions (i.e., by creating instances of these actions) that must occur in specific time intervals (referred by the propositional content of these constituting commitments).

Now, we can make clear that scheduled actions are neither action occurrences (i.e., particular events that occur in specific time intervals) nor action universals (patterns of features instantiated by multiple action occurrences). In fact, a scheduled action is not an action at all. Instead, it is a commitment to instantiate a specific action universal (plan) in a specific time interval, i.e., a closed appointment.

5 Analyzing and Improving a Software Process Ontology in terms of UFO

In [16], Falbo and Bertollo presented a Software Process Ontology that was developed for establishing a common conceptualization for software organizations to talk about software processes. This ontology is used as basis for the development of a process infrastructure for ODE [3], a Process-Centered Software Engineering Environment. Futhermore, in [16], this ontology is shown to be expressive enough to be used as a common ground for mapping the software process fragments of standards such as ISO/IEC 12207-ISO 9001:2000-ISO/IEC 15504, CMMI, RUP and SPEM. Figure 5 shows a fragment of this ontology.

As shown in figure 5, a *software process* can be decomposed into *activities* or other processes, called *sub-processes*. An *activity* is a piece of work that can produce *artifacts* (*outputs*). To be performed, an activity requires *resources*, adopts *procedures* and uses *artifacts* (*inputs*) produced by other activities. An activity can be decomposed into *sub-activities*, and can depend on the accomplishment of other activities, said *pre-activities*. Artifacts can be decomposed in *sub-artifacts*. *Resources* are used during, or to support, the execution of activities. These resources can be grouped into three main categories: (i) *human resource* are the roles that human agents is required to perform in an activity, such as requirement analyst, project manager, client and so on; (ii) *hardware resources* include any hardware equipment required to perform an activity, such as computers and printers; (iii) *software resources* concern any software product that is used in the accomplishment of an activity, such as a network management software or a data base management system. *Procedures* are adopted in the accomplishment of activities. There are several types of procedures. *Document Templates*, for instance, are models to be followed when preparing an artifact in an activity. A *method* is a systematic procedure that defines a *workflow* of activities (a set of steps) and heuristics to perform one or more activities. When a method can be adopted in the accomplishment of more than one activity, it has a workflow of activities for each one of them.

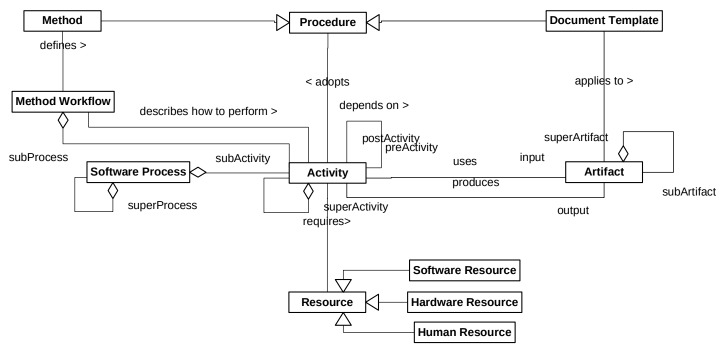

Fig.5. A Fragment of the ODE Software Process Ontology

Figure 6 represents a reengineered model of the Software Process Ontology by mappings its concepts to the fragments of the UFO-C ontology in figures 3 and 4. In figure 6, the concepts from UFO-C are shown in grey. By interpreting the ODE ontology in terms of UFO-C, it becomes clear that the former collapses the two notions of Action universal and Action Execution. We have performed this separation by introducing the terms Activity Occurrence (AO) and Software Process Occurrence (SPO) to denote particular actions that take place in specific time intervals. An activity occurrence can be atomic or complex. A SPO, in contrast, is necessarily a complex action. In fact, the distinction between a Software Sub-Process Occurrence (i.e., a SPO which is part of another SPO) and a Complex Activity Occurrence is not clear in the original ontology. Here, in order to eliminate this ambiguity, we assume that a SPO is the supreme of the composition lattice, i.e., a SPO is a complex action which is not part of any other complex event. A consequence of this definition is that there can be no subprocesses of a SPO and, thus, the reflexive parthood relation between SPO's has been removed from the model. In summary, a SPO is an instance of a Software Process which is in turn a subtype of a Complex Action Universal; an activity occurrence is an instance of an Activity which is a subtype of Action Universal (Plan).

In the software process ontology, an artifact is a type of Object (non-agentive substantial). The subArtifact/superArtifact relation between artifacts is thus governed by the mereological axioms defined for the (different types of) parthood relation between substantials defined in [1]. Although this cannot be elaborated here, we have that, generally, parthood between objects define an irreflexive, asymmetric and non-transitive relation; Parthood between processes is a strict partial order relation.

The notion of resource in the original ontology can also be mapped to the notion of Substantial in UFO-A, and the relation *requires* subsumes different modes of participating in an AO. We believe this issue should be elaborated in this domain ontology in order to make justice to the distinction between action contributions and (UFO-C) resource participations. In particular, a human resource (an Agent in UFO-C) cannot be *used*, *modified*, *created* or *terminated* by an AO. Instead, an action contribution of a human resource actually denotes a social commitment of that agent (with consequent permissions and obligations) of performing part of that AO. In terms of the formal relations we have put forth in [6], the *requires* relation for the case of human resources is a type of *dependence* relation between agents that will lead to a *delegation* relation when the process is instantiated or scheduled. In a nutshell, an agent *A* depends on an agent *B* iff *A* has a goal *G* that she cannot achieve by herself (either by lack of capacity or by the fact that *G* contrasts with one of her other goals), and *B* can achieve *G*. Agent *A* delegates goal *G* to *B* iff *A* depends on *B* for *G*; and *B* commits at achieving *G* for *A*. Software and Hardware resources are types of Objects and, thus, their modes of participations must be one of the types of (UFO-C) resource participation, namely, an usage participation. Notice, however, that whilst an output artifact is a (UFO-C) resource with a create participation in an AO, an input artifact is also a (UFO-C) resource with a usage participation. It is important to emphasize that a (UFO-C) resource is a *role* that an object plays in an event. As consequence, the same then must be the case for its subclasses: both artifact and resource must be roles played by objects in the scope of an AO. As a consequence, an object *O* that is an input artifact for an AO is an output artifact from another AO. Now, can an object *O* that is an output artifact for an AO be a software resource used by another AO? In other words, once human agents are not considered as resources, what characterizes the different roles output artifact and resource since both are played by objects with usage participations? The answer is the following: if an object plays the role of software resource in AO *ac* then there is no creating participation of this object in another AO *ac*' that is part of the same SPO.

A Procedure is a type of Normative Description. Two special types of Normative Descriptions in UFO-C are Object and Plan Descriptions. An Object Description is a description of an Object Universal (e.g., a blueprint of a house). In an analogous manner, a Plan Description is a description of a Plan (e.g., directives on how to assemble an IKEA chair). A method is a type of Plan Description; a Document Template is an Object Description.

Another issue in the original ontology is the relation of dependence between activities (now, activity occurrences). Dependence between activity occurrences is defined in terms of resources and, more specifically, of artifacts: an AO *a* depends on an AO *b* iff *a* uses as input a resource produced by *b*. However, the rolenames used for the relata of this relation are preActivity and posActivity indicating that there is an ordering relation hidden between these AO's. What temporal ordering relation(s) would this relation correspond to? It turns out it is not possible to extrapolate this answer from the data contained in the original ontology. In general, the only two temporal relations that can be eliminated from start are *finishes* and *equals*, i.e., if an AO *a* depends on something produced by an AO *b* then *a* cannot finish simultaneously or before *b*. Thus, considering ordering relations between instantaneous activity occurrences then dependence implies that *b* occurs *before a* (see UFO-B), since *starts* and *during* are not defined for instants and *meets* degenerates to *equals* in that case. However, for the case of time-extended AO, we are left with four possibilities: before, meets, starts and during. In any case, all these four relations are strict total orders (irreflexive, asymmetric and transitive) if we assume a linear model of time. In summary, since dependence between AOs is a total order plus a resource dependence, as a result we have that this relation must obey a partial order relation, i.e., an irreflexive, asymmetric and transitive but not necessarily totally defined relation between AOs.

The two relations between method workflow in the ontology of fig. 5 can be further clarified by making explicit the distinction between Action Universal and a description of that universal. A method workflow is an example of a description. Thus, it cannot be said to be composed of Activities. We take that, as a Plan description, a method workflow "describes how to perform" a complex action universal (a software process or a complex activity). This relation of "describe how to perform" is, hence, a specialization of the relation "describes" between Plan description and Plan. Moreover, as a description of a complex action universal, a method workflow describes how to perform the two or more actions that compose the instances of this universal.

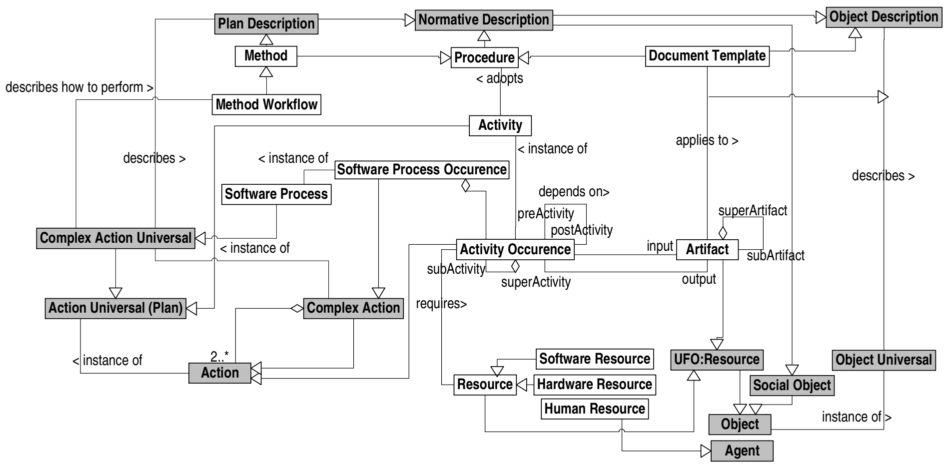

Fig.6. A Fragment of the Reengineered ODE Software Process Ontology

6 Final Considerations

In this paper, we presented the latest developments in the UFO foundational ontology. In particular, we discussed new versions of two fragments of UFO, namely UFO-B (concerned with *events*) and UFO-C (dealing with *social and intentional concepts*). In previous initiatives, UFO has been used to evaluate, re-design and integrate (meta) models of different conceptual modeling languages as well as to provide real-world semantics for their modeling constructs. In this paper, we illustrated the role played by philosophically grounded foundational ontology in the design of domain ontologies. In particular, we demonstrate how UFO can be used to evaluate, re-design and give real-world semantics to an ontology in the software engineering domain, namely, the software process ontology which is the core of the ODE Ontology-Centered Software Development Environment. By doing this, we have corrected a number of conceptual problems in this software engineering ontology by making it more truthful to the domain being represented and by making explicit its ontological commitments. Truthfulness to reality and conceptual clarity are fundamental quality attributes of conceptual models, in general, and of domain ontologies, in particular, and are directly responsible for the effectiveness of these models as reference frameworks for the tasks of reuse and semantic interoperability [1,5].

The foundational ontology as well as the domain ontologies discussed in this article are endowed with corresponding formal characterizations, which are not shown here, firstly, due to lack of space, but also since they are not essential to objectives of this paper. The Interested reader should refer to, for example, [1,16].

Acknowledgements This research has been partially supported by the TeleCardio Project (FAPES #31024866/2005), FAPES DCR Grant #37274554/2007 and INFRA-MODELA Project (FAPES/MCT/CNPq/CT-INFRA #019/2006).

References

1. Guizzardi, G. Ontological Foundations for Structural Conceptual Models, Universal Press, The Netherlands, 2005, ISBN 90-75176-81-3.
2. Falbo, R. A. et al., An Ontological Approach to Domain Engineering, Proceedings of the 14th SEKE, Ischia, Italy, ACM 2002.
3. Falbo, R.A., ODE: Ontology-based software Development Environment, IX Congreso Argentino de Ciencias de la Computación, La Plata, Argentina, Outubro 2003.
4. Guarino, N.; Guizzardi, G., In the Defense of Ontological Foundations for Conceptual Modeling, Scandinavian Journal of Inf. Systems, Vol.18, No. 1, ISSN 0905-0167, 2006.
5. Guizzardi, G., The Role of Foundational Ontology for Conceptual Modeling and Domain Ontology Representation, Proceedings of 7th DB&IS, Vilnius, IEEE Press 2006.
6. Guizzardi, R.S.S.; Guizzardi, G. Integrating Agent-Oriented Modeling Languages Using a Foundational Ontology, in Social Modeling for Requirements Engineering, P. Giorgini, N. Maiden, J. Mylopoulos, E. Yu (eds.), Cooperative Inf. Systems Series, MIT Press, 2008 (forthcoming).
7. Gärdenfors, P., Conceptual Spaces: the Geometry of Thought, USA, MIT Press 2000.
8. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A. ‘Ontology Library’, WonderWeb Deliverable D18, 2003.
9. Herre, H. et al, General Formal Ontology (GFO): A Foundational Ontology Integrating Objects and Processes. Technical Report Nr. 8. University of Leipzig, 2006.
10. Dockhorn, P., Guizzardi, G. et al., Situations in Conceptual Modeling of Context, Proceedings of the 2nd VORTE, Hong Kong, IEEE Digital Library 2006.
11. Allen, J.F., Maintaining Knowledge About Temporal Intervals, Communications of the ACM, Vol.26, no. 11, 1983.
12. Bottazzi, E.; Ferrario, E., Towards a DOLCE Ontology of Organizations, Journal of Business Process Integration and Management (IJBPIM), Inderscience Pub., 2008 (forth).
13. Searle, J., Mind, Language and Society, Basic Books, 2000.
14. Conte, R., Castelfranchi, C., Cognitive and Social Action, UCL Press, London, 1995.
15. Stanford Encyc. of Philosophy, Action, http://plato.stanford.edu/entries/action/, 2002.
16. Falbo, R.; Bertollo, G., Establishing a Common Vocabulary for Helping Organizations to Understand Software Processes, Proceedings of the 1st VORTE, Enschede, The Netherlands, ISSN 1574-0846, 2005.