# Using a Foundational Ontology for Reengineering a Software Process Ontology

Ana C. O. Bringuente, Ricardo A. Falbo, Giancarlo Guizzardi

Universidade Federal do Espírito Santo, Brazil
{acobringuente,falbo,gguizardi}@inf.ufes.br

**Abstract.** During project planning, knowledge about software processes is useful in several situations: software processes are defined, activities are scheduled, and people are allocated to these activities. In this context, standard software processes are used as basis for defining project processes, and tools are used to support scheduling, people allocation, and so on. Ideally, people and tools should share a common conceptualization regarding this domain for allowing interoperability, and correct use of the tools. A domain ontology can be used to define an explicit representation of this shared conceptualization. Moreover, for a domain ontology to adequately serve as a reference model, it should be built explicitly taking foundational concepts into account. This paper discusses the reengineering of part of a Software Process Ontology based on the Unified Foundational Ontology (UFO). The part reengineered concerns standard processes, project processes, and activities, which are analyzed at the light of UFO concepts.

Categories and Subject Descriptors: D.2.13 [**Software Engineering**]: Reusable Software—*Domain engineering, Reuse models*

Keywords: Domain Ontology, Foundational Ontology, Ontology Reengineering, Software Process, Unified Foundational Ontology

## 1. INTRODUCTION

Managing a project involves applying knowledge, skills, tools and techniques to project activities in order to meet its requirements. Such knowledge application requires the effective management of suitable processes [PMI 2008]. Among these processes, we can highlight the importance of the Project Management Process, which is responsible, among others, for defining activities, scheduling them, and allocating human resources for each planned activity.

Regarding software projects, project management is a critical process. During project planning, software processes should be defined, their activities should be scheduled, and people should be allocated to them. In this context, standard software processes are used as basis for defining project processes. Once the processes are defined, tools are used to support scheduling, team allocation, and so on. During project execution, activities are performed and people accomplish their tasks, recording the time they spent on them.

Generally, different tools are used to support such tasks. Considering that these tasks are iterative and interrelated, ideally these tools should interoperate. On the other hand, in order to correctly use the tools, developers and tools should share a common conceptualization regarding the domain of software processes. In fact, for both purposes of interoperability aforementioned, we need a shared conceptualization regarding the software process domain. In this context, a domain ontology can be used to define an explicit representation of this shared conceptualization.

This research is funded by the Brazilian Research Funding Agencies FAPES (Process Number 45444080/09) and CNPq (Process Numbers 481906/2009-6 and 483383/2010-4).

For being able to adequately serve as a reference model, a domain ontology should be constructed using an approach that explicitly takes foundational concepts into account. The use of foundational concepts that take truly ontological issues seriously is becoming more and more accepted by the ontological engineering community, especially for representing complex domains [Guizzardi et al. 2010], such as is the case of software processes. A reference ontology is developed with the aim at representing the subject domain with truthfulness, clarity and expressivity, regardless of computational requirements. The main goal is to help modelers externalize their knowledge about the domain, to make their ontological commitments explicit to support meaning negotiation, and to improve the tasks of domain communication, learning and problem solving [Guizzardi et al. 2010].

This paper discusses the reengineering of part of the Software Process Ontology (SPO) originally proposed in [Falbo and Bertollo 2009],based on the Unified Foundational Ontology (UFO) [Guizzardi 2005; Guizzardi et al. 2008]. This ontology was partially reengineered at the light of UFO in [Guizzardi et al. 2008]. This first reengineering initiative focused on distinguishing between process/activity and process/activity occurrence, among others. In this paper, we still follow the approach adopted in [Guizzardi et al. 2008],i.e., to align the SPO concepts and relations to the concepts and relations of UFO. However, we focus on improving these distinctions analyzing concepts such as standard software processes, project processes and activities, at the light of UFO concepts of events, commitments, appointments and normative descriptions. Our focus is directed towards concepts involved in management activities that are related to process definition, scheduling and resource allocation, since we intend to use the reengineered version of this ontology as basis for integrating tools supporting these activities. Thus, other parts of the SPO were not analyzed and reengineered in this work.

The paper is organized as follows. Section 2 talks briefly about the domain of interest: software process and project planning. Section 3 presents the concepts of UFO that are relevant to this work. Section 4 discusses the first initiative in reengineering the SPO [Guizzardi et al. 2008]. Section 5 presents the advances in reengineering the SPO. Section 6 discusses some related works, and, finally, Section 7 presents our conclusions.

## 2. PROJECT AND PLANNING SOFTWARE PROCESS

According to ISO 10006:2003, Project Management, in general, includes planning, organization, supervision and control of all aspects of the project in a continuous process to achieve their goals. A desired result is achieved more efficiently when the project activities and related resources are managed as a process [ISO 2003]. The Project Management Processes are used to establish and evolve project plans, to assess actual achievement and progress against the plans and to control execution of the project through to fulfillment [ISO/IEC 2008].

Processes are decomposed into smaller pieces. In the context of software engineering, some processes are decomposed into activities and others into lower-level processes. A lower-level process is described when the decomposed portion of the process itself satisfies the criteria to be a process, i.e., it has a purpose, is cohesive, and it is placed under the responsibility of an organization or a party in the software life cycle. An activity is used when the decomposed unit does not qualify as a process and can be considered as simply a collection of tasks [ISO/IEC 2008]. The project processes should be identified and documented, and the activities should be carried out and controlled in accordance with the project plan [ISO 2003]. Software project processes should be defined considering the activities to be accomplished, the required resources, the input and output artifacts, the adopted procedures (such as methods, techniques, templates) and the life cycle model to be used [Falbo and Bertollo 2009].

During project process definition, the organization should take account of the experiences gained in developing and using its processes. This may be accomplished by identifying the appropriate processes for the project, identifying inputs, output and objectives for the project's processes, and assigning authority and responsibilities for the processes, among others [ISO/IEC 2008]. Although different

projects require processes with specific features, it is possible to establish a set of software process assets that should be present in all project processes of an organization. This set of process assets is called an organizational standard software process. The organization's set of standard processes is tailored by projects to create their defined processes. Other organizational process assets are used to support tailoring and implementing defined processes. A standard process is composed of other processes (i.e., subprocesses) or process elements that describe activities and tasks to consistently perform the work [SEI 2010]. This tendency in using standard processes is advocated by almost every process quality models and standards, including ISO/IEC 12207 [ISO/IEC 2008] and CMMI [SEI 2010]. All of them suggest the use of a standard process as the starting point from which project's processes can be defined.

The main goal of project planning is to define a scope for the project, refine the goals and develop the necessary actions to achieve them [PMI 2008]. So, the Project Planning Process should [SEI 2010]: (i) determine the scope of the project and technical activities, (ii) identify process outputs, project tasks and deliverables, (iii) establish schedules for project task conduct, including achievement criteria and required resources to accomplish project tasks. It involves, among others, determining effort estimation, activities and tasks to be done, adequate resources needed to execute these tasks, schedules for the timely completion of tasks, allocation of tasks, and assignment of responsibilities.

In the context of project management, resource-related processes aim to plan and control resources, including equipments, facilities, materials, software, services, personnel and space. Personnel in the project organization should have well-defined responsibility and authority for their participation in the project. The authority delegated to the project participants should correspond to their assigned responsibilities. The quality and success of a project will depend on the participating personnel. Therefore, special attention should be given to the activities in the personnel-related processes, such as the allocation of personnel. When assigning members to project teams, their interests, strengths and weaknesses should be considered. The job or role to be played should be understood and accepted by the person assigned [ISO 2003].

Time-related processes aim to determine dependencies and the duration of the activities, and to ensure timely completion of the project. They include planning activity dependencies, estimation of duration, and schedule development and control. Scheduling can impact on the resources of the project. In fact, time-related processes and resource-related processes (including personnel-related processes) are very interrelated. Decisions or actions taken in the context of one of them should be carefully analyzed, taking into account their implications for the other processes [ISO 2003].

## 3. THE UNIFIED FOUNDATIONAL ONTOLOGY - UFO

UFO is a foundational ontology that has been developed based on a number of theories from Formal Ontology, Philosophical Logics, Philosophy of Language, Linguistics and Cognitive Psychology. It is composed by three main parts. UFO-A is an ontology of endurants [Guizzardi 2005]. UFO-B is an ontology of perdurants (events). UFO-C is an ontology of social entities (both endurants and perdurants) built on the top of UFO-A and UFO-B [Guizzardi et al. 2008]. A complete description of UFO falls outside the scope of this paper. However, in the sequel we describe some of its concepts that are important for this paper. This description is based mainly on [Guizzardi et al. 2008].

As shown in Figure 1, a fundamental distinction in UFO-A is between particulars and universals. ***Particulars*** are entities that exist in reality possessing a unique identity, while ***universals*** are patterns of features, which can be realized in a number of different particulars. ***Substantials*** are existentially independent particulars. ***Moments***, in contrast, are particulars that can only exist in other particulars, and thus they are dependent of them. In this sense, moments are existentially dependent on other particulars. Existential dependence can also be used to differentiate intrinsic and relational moments: ***intrinsic moments*** are dependent of one single individual (e.g., a color,

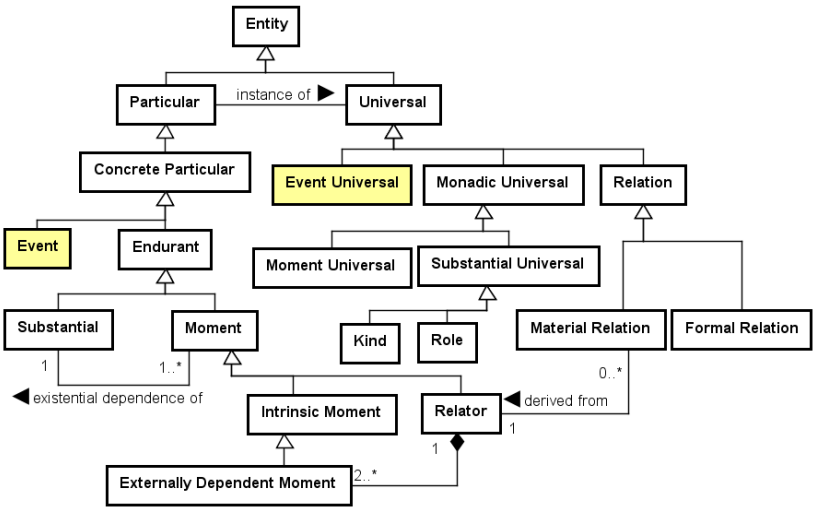


Fig. 1. A Fragment of UFO-A – An Ontology of Endurants

a temperature), while ***relators*** depend on a plurality of individuals (e.g., an employment, a medical treatment, a marriage). Most distinctions made for particulars apply to universals. Thus, we have the counterparts Substantial Universal, Moment Universal, Intrinsic Moment Universal and Relator Universal, although the last two were not shown in Figure 1.

Regarding substantial universals, while persisting in time, substantial particulars can instantiate several substantial universals. Some of these types, a substantial instantiates necessarily (i.e., in every possible situation) and define what the substantial is. These are the types named ***kind***. There are, however, types that a substantial instantiates in some circumstances, but not in others, such as is the case of roles. A ***role*** is a type instantiated in a given context, such as the context of a given event participation or a given relation (e.g., student). Both kind and role are ***sortal substantial universals***, but kind is a ***rigid sortal***, while role is an ***anti-rigid sortal***. Although not represented in Figure 1, ***Sortal Universal***, ***Rigid Sortal*** and ***Anti-rigid Sortal*** are concepts of UFO-A. For details see [Guizzardi 2005].

***Relations*** are entities that glue together other entities. ***Formal relations*** hold between two or more entities directly, without any further intervening individual. ***Material relations***, conversely, have material structure of their own. In other words, for a material relation to exist, a ***relator*** that ***mediates*** the related entities must exist. Thus, relators are particulars with the power of connecting entities. The relations between relators and the connected entities are said ***mediation relations***.

UFO-B makes a distinction between enduring and perduring particulars (endurants and perdurants). ***Endurants*** are said to be wholly present whenever they are present, i.e., they are in time, (e.g., a house). Perdurants or ***Events***, in contrast, are particulars composed of temporal parts, i.e., they happen in time in the sense that they extend in time accumulating temporal parts (e.g., a party). The concepts ***Event*** and ***Event Universal***, because they are not part of UFO-A, were shown detached in Figure 1. Figure 2 depicts a fragment of UFO-B.

The main category on this ontology is ***Event***. Events can be Atomic or Complex. ***Atomic events*** have no improper parts, while ***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. Moreover, since events happen in time, they are framed by a time interval. The model of Figure 2 depicts these two aspects on which events can be analyzed,

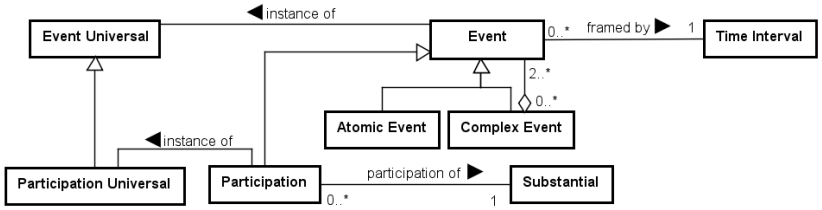


Fig. 2. A Fragment of UFO-B – An Ontology of Events

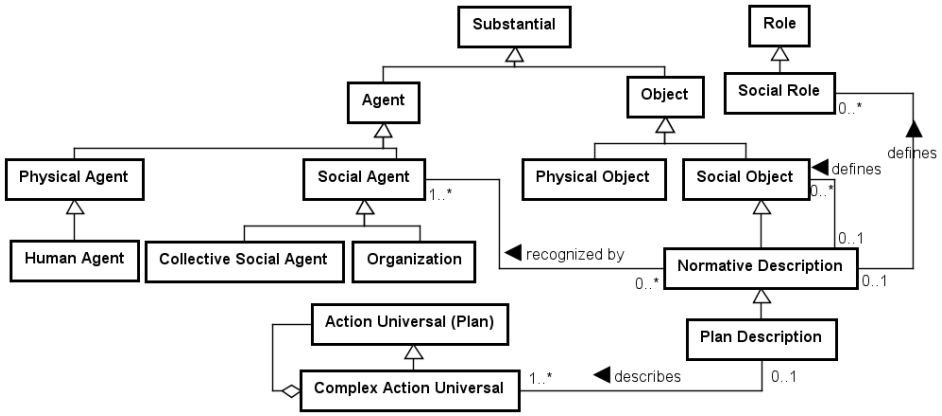


Fig. 3. A Fragment of UFO-C: Distinction between Agent and Object

namely, as time extended entities with certain (atomic or complex) mereological structures, and as ontologically dependent entities which can comprise of a number of individual participations.

The third layer of UFO is an ontology of social entities (both endurants and perdurants). As shown in Figure 3, one of the main distinctions made in UFO-C is between agents and non-agentive objects. An ***agent*** is a substantial that creates actions, perceives events and to which we can ascribe mental states (intentional moments). Agents can be physical (e.g., a person) or social (e.g., an organization). A ***human agent*** is a type of ***physical agent***. An ***object***, on the other hand, is a substantial unable to perceive events or to have intentional moments. Objects can also be further categorized into physical (e.g., a book, a car) and social objects (e.g., money, language). A ***normative description*** is a type of ***social object*** that defines one or more rules/norms recognized by at least one social agent and that can define nominal universals such as social objects, social roles and descriptions of plans (action universals). A ***plan description*** is a special type of normative descriptions that describes complex action universals (complex plans).

Agents are substantials that can bear special kinds of moments named ***Intentional Moments***. Intentional moments can be ***social moments*** or ***mental moments***. Intentionality in UFO should be understood in a broader sense than the notion of "intending something". It refers to the capacity of some properties of certain individuals to refer to possible situations of reality. Thus, "intending something" is a specific type of intentionality termed Intention in UFO. ***Intentions*** (or internal commitments) are mental moments that represent an internal commitment of the agent to act towards that will. They cause the agent to perform actions. The propositional content of an intention is a ***goal***. Figure 4 shows a fragment of UFO-C dealing with these concepts.

Besides internal commitments (intentions), there are also social commitments. A ***social commitment*** is a commitment of an agent A towards another agent B. As an externally dependent moment, a social commitment inheres in A and is externally dependent on B. The social commitments necessarily cause the creation of an internal commitment in A. Also, associated to this internal commitment, a

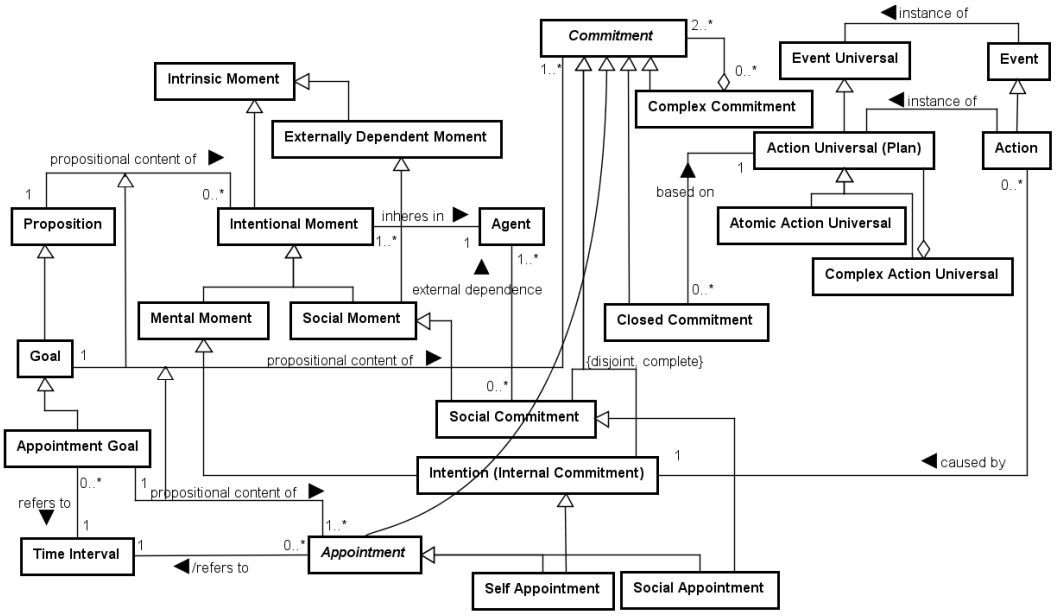


Fig. 4. A Fragment of UFO-C: Commitments and Appointments

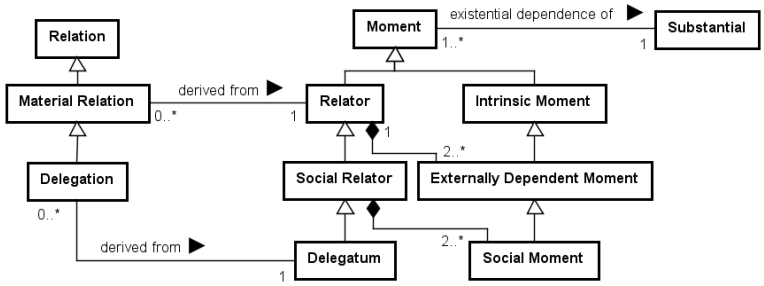


Fig. 5. A Fragment of UFO-C: Delegation

social claim of B towards A is created. Commitments and claims always form a pair that refers to a unique propositional content, and a ***social relator*** is an example of a relator composed of two or more pairs of associated commitments/claim. For sake of simplicity, these aspects are not shown in Figure 4. See [Guizzardi et al. 2008] for details.

Commitments can be atomic or complex. ***Complex commitments*** are composed of other commitments. A ***closed commitment*** is a commitment that is based on an Action Universal (Plan). The closed commitment is fulfilled by the agent only if the agent satisfies the commitment goal by executing an action which is an instance of the plan. A special type of commitment is an appointment. An ***appointment*** is a commitment whose goal explicitly refers to a time interval. Like commitments, appointments can be either Internal (***Self-Appointment***) or ***Social Appointments***. Moreover, a ***closed appointment*** is a closed commitment whose goal explicitly refers to a time interval.

A special type of material relation is delegation. ***Delegation*** is a material relation derived from a social relator ***delegatum***. When an agent A (called delegator) delegates a goal to an agent B (called delegatee), a social claim of B towards A is created. By B assuming the achievement of the goal on behalf of A, a social commitment of B towards A is also created. The commitment/claim pair composes the delegatum from which the delegation is derived from, as shown in Figure 5.

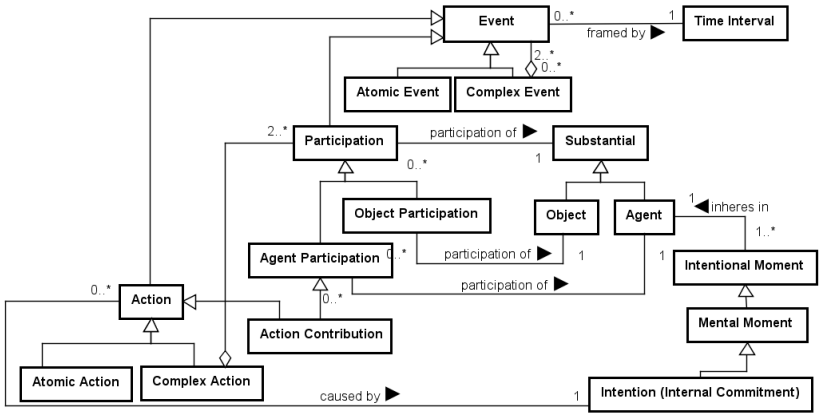


Fig. 6. A Fragment of UFO-C: Actions and Participations

Finally, ***actions*** are intentional events, i.e., they have the specific purpose of satisfying some intention (e.g., a business process, a communicative act). 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. It is worthwhile to point out 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 does not have intention. Figure 6 shows a fragment of UFO-C dealing with these concepts.

## 4. THE SOFTWARE PROCESS ONTOLOGY

As discussed in Section 2, during project planning, we need to define the project process, schedule its activities, and allocate people to perform them. For purposes of project monitoring and control, we need to track the accomplishment of the activities, and the time spent to perform them.

During the definition of a software process for the project, the project manager should identify the activities that have to be performed in order to achieve the project goals. This is done by tailoring organizational standard processes, taking the project particularities and team features into account. The project process is the basis for the further project management activities. After defining the process, the project manager has to create a network of activities, define how long each activity will last, and allocate people to perform it. For a good understanding of these tasks, we need a shared conceptualization regarding software processes.

The Software Process Ontology (SPO) originally developed in [Falbo and Bertollo 2009] was built aiming at establishing a common conceptualization for software organizations to talk about software process. It was divided into four sub-ontologies, namely: activity, resource, procedure and software process ontologies. Figure 7 shows a fragment of its first version that includes concepts from the activity, resource, and software process sub-ontologies. We chose this fragment, because in this paper we are interested in the portion of the SPO conceptualization that is more relevant for project planning.

According to [Falbo and Bertollo 2009], the model depicted in Figure 7 aims to represent the following universe of discourse. A *software process* can be decomposed into activities or other processes, called sub-processes. An *activity* is a piece of work that requires *resources* (humans, software and hardware) to be performed. An *activity* can be decomposed into sub-activities, and can depend on the accomplishment of other activities, said pre-activities. *Resources* are things required to the accomplishment of an activity, such as people, hardware and software. A *standard process* is a generic process institutionalized in an *organization*, establishing basic requirements for project processes to

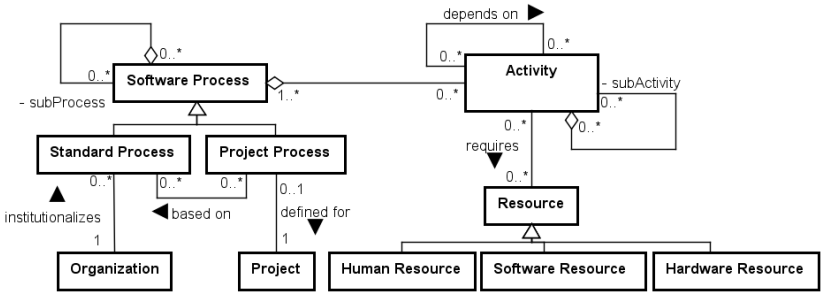


Fig. 7. A fragment of the Software Process Ontology (Original Version)

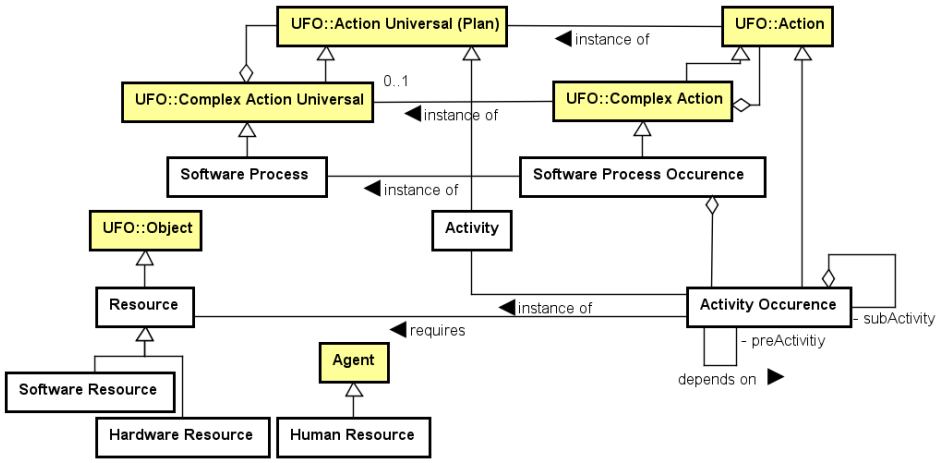


Fig. 8. A Fragment of the SPO reengineered in [Guizzardi et al. 2008]

be performed in that organization. A *project process* refers to the process defined for a specific *project*, considering the particularities of that project.

As said before, the original version of the SPO was object of a partial reengineering in [Guizzardi et al. 2008] by mappings some of its concepts to concepts of UFO. Figure 8 shows a fragment of this reengineered version of SPO, showing concepts from UFO detached. Again, we chose this fragment because our focus is on project planning. By interpreting the original version of the SPO in terms of UFO, Guizzardi, Falbo and Guizzardi [Guizzardi et al. 2008] pointed out that the former collapses the notions of action universals and actions. To solve this problem, they introduced the concepts of Activity Occurrence and Software Process Occurrence to denote particular actions that take place in specific time intervals. Moreover, they said that a software process occurrence is an instance of a software process, which is, in turn, a Complex Action Universal. Analogously, an activity occurrence is an instance of an activity, which is an Action Universal (Plan).

The notion of resource in the original version of SPO was mapped to the notion of (Non-Agentive) Object in UFO. The concept of human resource was no longer considered a resource, but is considered a type of Agent. This opened space for distinguishing object participations from agent participations. Thus the relation requires in the original version of SPO subsumes different modes of participating in an activity occurrence, namely object participations and action contributions. 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 activity occurrence. Thus, 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. Others changes were made, but they are out

of the scope of this paper. For more information, see [Guizzardi et al. 2008].

Although an important ontological analysis was done in [Guizzardi et al. 2008], there are many aspects that remained open. What is a standard process? What is the difference from an activity that is defined in the project process and the one that is scheduled and performed? How can we treat people allocation? To answer these and other questions, we go a step ahead in the SPO reengineered process. The next section presents the results we have already achieved.

## 5. A STEP AHEAD IN REENGINEERING THE SOFTWARE PROCESS ONTOLOGY

In the first initiative of reengineering the SPO, some important issues were not addressed, namely: (i) the distinction between standard software processes and project processes; (ii) the distinction between planned but not scheduled activities, scheduled activities, and activity occurrences. In this paper, the main purpose is advancing in these aspects, as well as addressing the problem of allocating people to perform the scheduled activities. In the sequel, we elaborate in these distinctions.

### 5.1 Standard Process

Although the original version of the SPO distinguishes between standard software processes and project processes, the reengineering performed in [Guizzardi et al. 2008] did not take this distinction into account. As discussed in the previous section, a *standard process* refers to a generic process defined by an organization, establishing basic requirements for processes to be performed in that organization. Looking to the conceptualization established by UFO, a *standard process* should be viewed as a ***Complex Action Universal***, described by a ***Plan Description***. Recognizing that, we introduced the concept of *standard process definition document* as the plan description that describes the standard process. As a normative description, the *standard process definition document* defines rules recognized by the organization for conducting its projects.

According to the domain (see Section 2), a *general standard process* is a standard process that is composed of *specific standard processes*, allowing an organization to define sub-processes of a standard process. For instance, the organizational standard process of an organization can be decomposed into standard sub-processes for software development and project management. A *specific standard process* (e.g., a standard Project Management Process), in turn, is decomposed into *standard activities*, which are ***Action Universals***. As action universals, *standard activities* can be atomic or complex.

Figure 9 introduces these concepts in the new version of SPO. Concepts from UFO are shown detached, as well as the subtype relationships that map concepts and relations from SPO to UFO. It is worthwhile to point out that the way of structuring standard processes and standard activities is perceived in the domain of software processes. It is not directly inspired in UFO, although it is aligned with UFO. In this work we attempted to be more precise than in the first reengineering effort [Guizzardi et al. 2008] regarding the relation's multiplicities. Aspects such as weak supplementation, for instance, have been considered. Thus, we stated that a general standard process should be composed of at least two specific standard processes. The same applies to complex standard activities.

### 5.2 Project Process Definition and Scheduling

In contrast with standard processes, *project processes* refer to the processes defined for specific *projects*, considering the particularities of these projects. When a project manager defines a project process, he/she is performing a communicative act that creates an ***internal commitment*** (an ***Intention*** in UFO) of the organization: the commitment of performing the activities defined in the project process. Thus, in terms of UFO, a project process is a type of ***Intention***, as shown in Figure 10.

A project process is also a complex commitment, since it is composed by other commitments. Analogously to standard processes, a *general project process* is a project process that is composed of

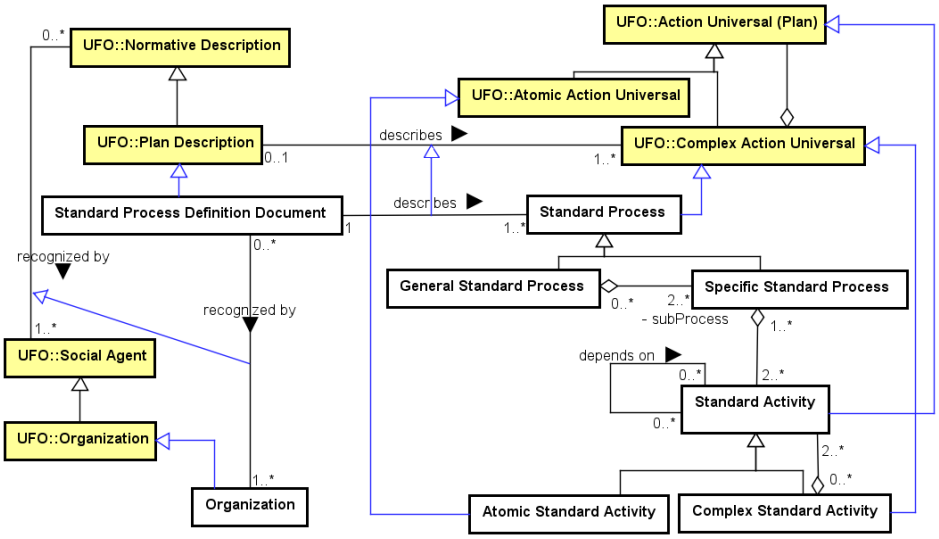


Fig. 9. Standard Process as a Complex Action Universal described by a Plan Description

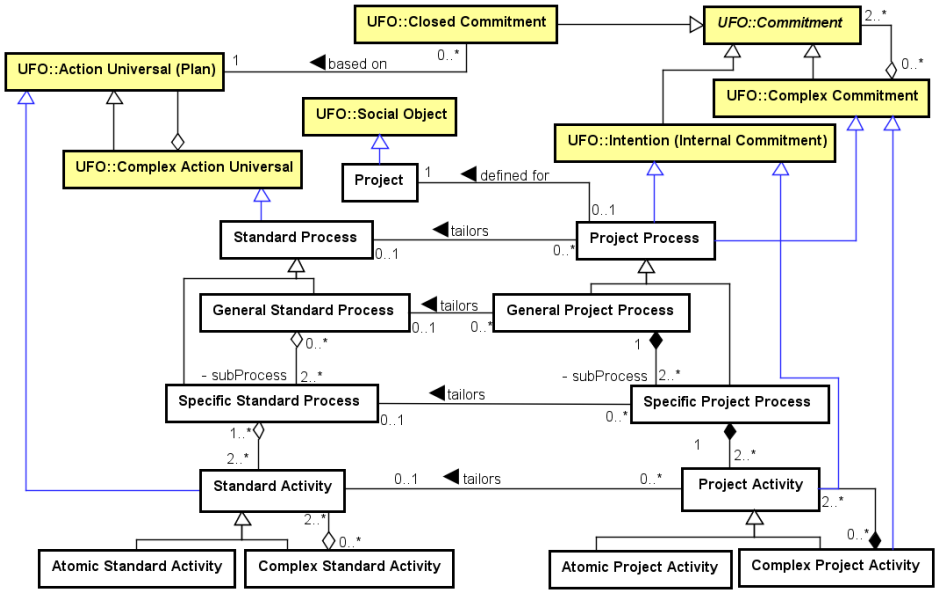


Fig. 10. Project Processes and Activities as Internal Commitments of the Organization

*specific project processes*, allowing an organization to define sub-processes that are part of a general project process. A *specific project process*, in turn, is decomposed into *project activities*, which are also ***internal commitments*** (***Intention***) of the organization and can be atomic or complex. Project processes can be defined by tailoring standard processes. Analogously, project activities can be defined by tailoring standard activities. When the project process follows the standard process without adding or changing its assets, we can say that it is a ***closed commitment***, since the organization must fulfil the commitment by executing a specific action. The same applies to *project activities*.

Once defined the project process, its activities can be scheduled. Scheduling refers to defining start and end dates for each process/activity. In other words, the time intervals in which the project processes and activities should occur are defined. In terms of UFO, this corresponds to ***appointments***,

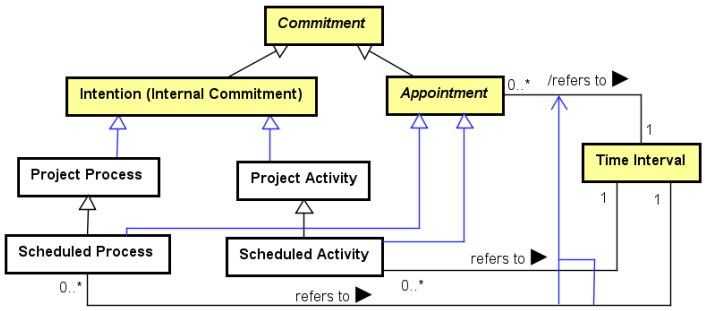


Fig. 11. Scheduled Processes and Activities as Appointments

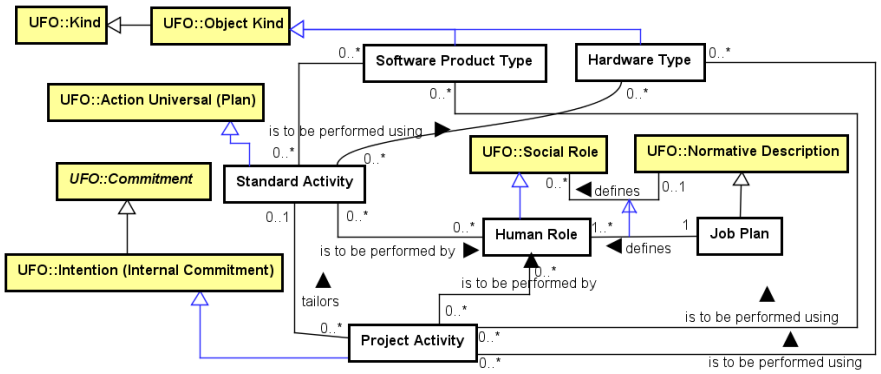


Fig. 12. Software Product Type and Hardware Type as Object Kind; Human Role as a Social Role

i.e., commitments that explicitly refer to a time interval, as shown in Figure 11.

## 5.3 Resource Types

*Standard activities* indicate the *software product types* (e.g., a UML modeling tool) and the *hardware types* (e.g., a notebook) to be used, and the *human roles* (e.g. requirements engineer) that shall perform activities of such type. The same occurs when defining project processes: *project activities* indicate the *software product types* and *hardware types* to be used and the human role people should play when performing the actions (*activity occurrences*) for satisfying such intentions [1].

Software product types and hardware types are object kinds in UFO, as shown in Figure 12, while human roles are social roles in UFO, defined by a normative description (in the case, a job plan) recognized by the organization. It is worthwhile to point out that in the definition of a standard process or even in the definition of a project process, we are not defining which specific software product or hardware is to be used, but only their types (object kinds, in UFO), and thus we are not talking about resources, but about resource types. The same applies to human resources. Again, we are not allocating people, but only defining the social roles that people should play when performing the corresponding activity occurrences (see subsection 5.5).

[1]Remember that project activities are ***intentions*** in UFO and, thus, ***actions*** (activity occurrences, in SPO) are performed for satisfying such ***intentions***.

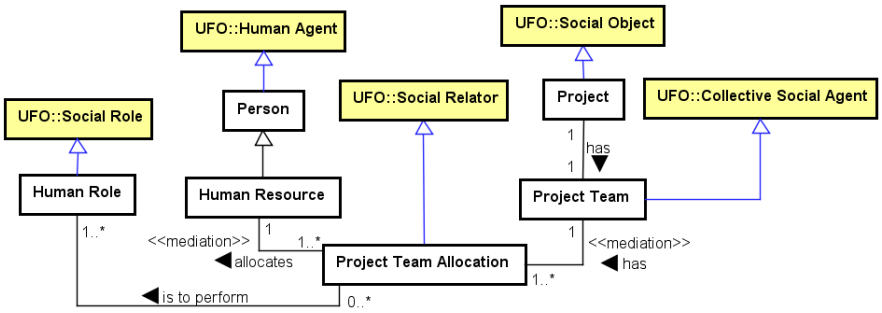


Fig. 13. Team Allocation as Social Relators

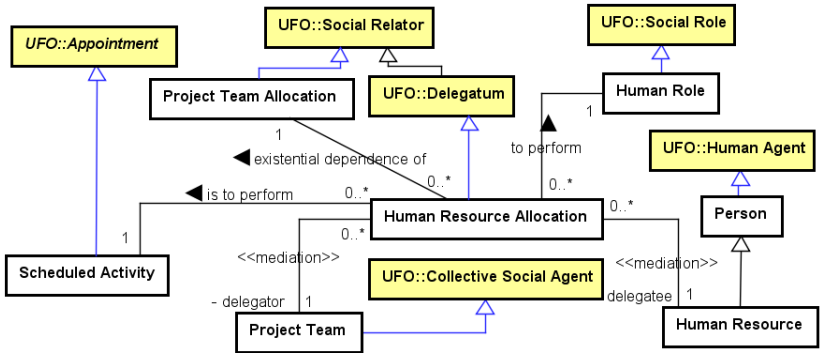


Fig. 14. Human Resource Allocation as Delegatum

## 5.4 Human Resource Allocation

As pointed in Section 2, for a project to succeed, special attention should be given to personnel-related tasks, such as assigning members to project teams and allocating them to perform project activities. A person is assigned to a project team for performing certain human roles. Thus, according to UFO, *project team allocation* (allocation of a person to a project team) is a type of ***Social Relator***, which is composed of a ***commitment*** of the person to the team in performing according to the human roles assigned to her/him, and a ***claim*** of the team towards the person. Moreover, the person is now a *human resource* for the project, since she/he can be allocated to the project activities. In terms of UFO, *person* is a type of ***Human Agent***, while *project team* is a type of ***Collective Social Agent***. Figure 13 shows these concepts.

Once we have assigned members to the *project team*, we can allocate these *human resources* to perform *scheduled activities*. In this case, the *project team* is delegating responsibilities to *human resources* and thus *human resource allocation* is a type of ***delegatum*** in UFO, in which the *project team* is the ***delegator*** and the *human resource* is the ***delegatee***. The team is delegating to the human resource the responsibility of performing an action to satisfy the goal of the scheduled activity, by performing a specific human role. In other words, the delagatum *Human Resource Allocation* derives a material relation between *human resource* and *project team*, where the *project team* acts as a ***delegator***, while the *human resource* acts as a ***delegatee*** aiming at achieving the ***appointment goal*** of the *Scheduled Activity*. Figure 14 shows the model capturing this conceptualization. It is important to emphasize that there is a relation of ***existential dependence*** between *Human Resource Allocation* and *Project Team Allocation*, indicating that, for a human resource allocation to exist (for a human resource be allocated to a scheduled activity), she/he has to be first allocated to the project team. For sake of simplicity, goals are not shown in Figure 14.

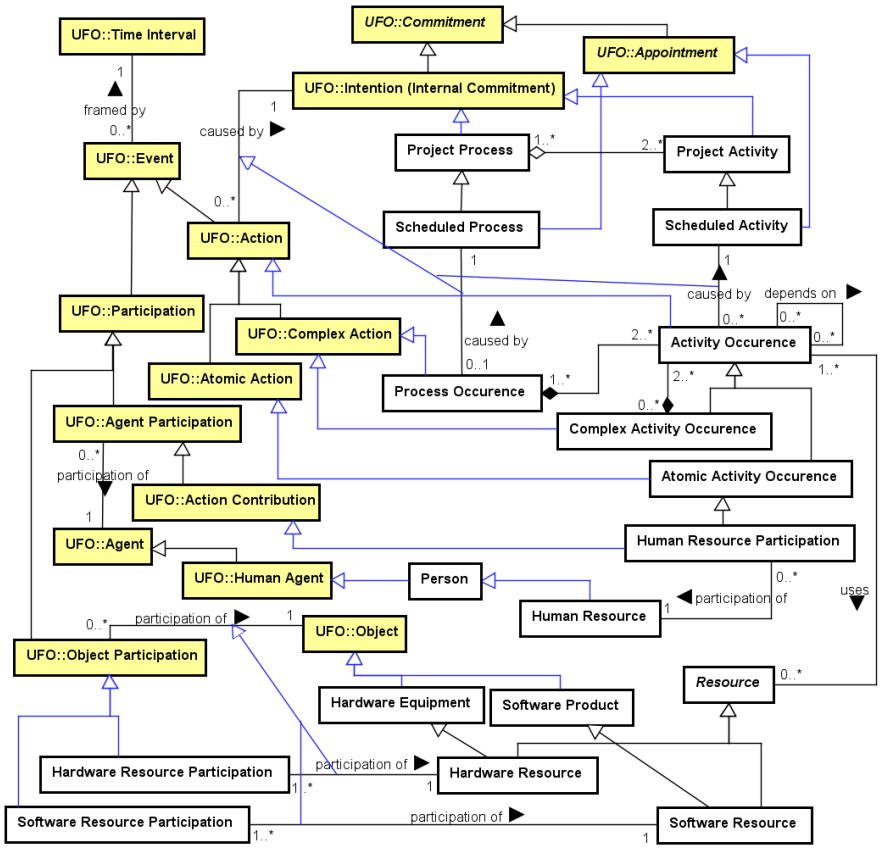


Fig. 15. Process and Activity Occurrences as Actions

## 5.5 Activity Occurrence

Finally, actions are performed in the context of the project, in order to achieve its objectives. As pointed in [Guizzardi et al. 2008], process and activity occurrences are effectively the ***actions*** in the sense of UFO. ***Actions*** are intentional events, i.e., they have the specific purpose of satisfying some intention. In other words, ***actions*** are caused by ***intentions***. *Process occurrences* are ***Complex Actions*** caused by *scheduled processes*, while *activity occurrences* are ***Actions*** caused by *scheduled activities*. As shown in Figure 15, both scheduled processes and scheduled activities are intentions, and the relations "caused by" between process occurrence and scheduled process and between activity occurrence and scheduled activity are types of the relation ***caused by*** between ***actions*** and ***commitments*** in UFO.

Like project activities, activity occurrences can be complex or atomic. In this case, *complex activity occurrences* are ***Complex Actions***, while *atomic activity occurrences* are ***Atomic Actions***. Moreover, we can distinguish human resource participations. A *human resource participation* is the participation of a specific *human resource* in a complex activity occurrence. Since this participation is an intentional one, then *human resource participations* are ***Action Contributions***. This concept is very important for the purpose of controlling the project. Almost all organizations require that their human resources register the time they spent in doing a task. Human resource participations, as ***events***, are framed by a ***time interval***, capturing this important aspect. In contrast, hardware and software resource participations are non intentional participations of ***objects*** (*hardware equipments* and *software products*) and thus they are ***object participations*** in UFO's sense. Moreover, a *hardware equipment* plays the role of a *hardware resource* when it is used in an *activity occurrence*. The

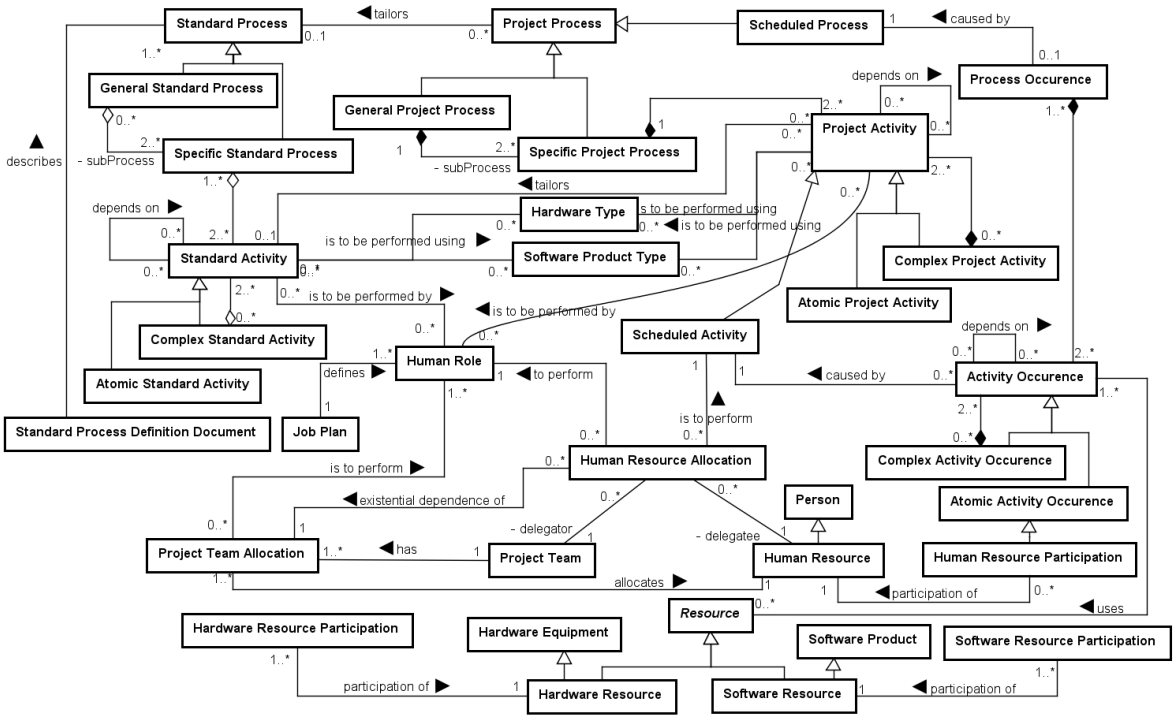


Fig. 16. The current version of the Software Process Ontology

same applies to a *software product*.

The new version of the Software Process Ontology (Fig. 16), obtained as a result from our reengineering efforts. Contrasting it to the original version (Fig. 7) or to the first reengineered version (Fig. 8) of the same ontology, we can see that there was much knowledge missing. This corroborates the fact that using UFO as basis for ontology reengineering helped us to externalize the knowledge about the domain, making the ontological commitments explicit, as pointed in [Guizzardi et al. 2010].

## 6. RELATED WORKS

There are some works that have proposed ontologies for the software process domain. This is the case of [Falbo and Bertollo 2009], analyzed in this paper, and [Liao et al. 2005]. In the latter, an OWL-based ontology is proposed for software processes. As an OWL ontology, it is, in fact, an implementation of an ontology and, thus, it is full of encoding bias. As pointed out by Thomas Gruber [Gruber 1995], an ontology should be specified without depending on a particular symbol-level encoding. Encoding bias results from choosing a representation purely for the convenience of implementation, and should be minimized. In this paper, we are interested in a reference domain ontology, as defined by Gizzardi [Guizzardi et al. 2010] as an ontology developed with the aim at representing the subject domain with truthfulness, clarity and expressivity, regardless of computational requirements. Concerning the conceptualization of the Liao's and colleague's ontology, it is too simple and focuses on processes and practices, where processes can be basic and composite, while practices can be basic or atomic. Moreover, a process is defined by an organization and belongs to a process category. However, like [Falbo and Bertollo 2009], it is not clear what a process or a practice means. Is a process/practice a standard process/activity, a project process/activity or an occurrence? This is not explicitly stated in the ontology. We should highlight that we are interested in building a reference domain ontology, and thus we are looking for software process ontologies developed using an approach that explicitly takes foundational concepts into account. At the best of our knowledge, this is only the case of [Guizzardi

et al. 2008], whose results were the base for the improvements done in this paper.

Regarding studies dealing with the evaluation and improvement of conceptual models based on foundational ontologies, recently some have been developed, specially using UFO [Guizzardi et al. 2010]. In the software engineering domain, Barcellos and Falbo [Barcellos and Falbo 2009] reengineered a Software Enterprise Ontology based on UFO. This is also the case of [Falbo and Nardi 2008], where Falbo and Nardi developed an Ontology of Software Requirements. Finally, in [Barcellos et al. 2010], Barcellos, Falbo and Dal Moro developed a Measurement Ontology. In all cases, these ontologies were developed or reengineered taking concepts and relations of UFO into account, as in this work.

However, there are other approaches that are not based on UFO. Guarino and Welty [Guarino and Welty 2002] developed the OntoClean methodology. OntoClean aims at providing guidance on which kinds of ontological decisions need to be made, and on how these decisions can be evaluated based on general ontological notions drawn from philosophical ontology. In [Welty et al. 2004], Welty, Mahindru and Chu-Carroll report the results of experiments that measure the advantages achieved from the use of ontologies improved based on OntoClean. Finally, Silva et al. [Silva et al. 2008] applied a technique that is based on OntoClean, called VERONTO (ONTOlogical VERification), to improve analysis patterns in the geographic domain.

As OntoClean, UFO is being used to evaluate, re-design and give real-world semantics to domain ontologies. However, since UFO's conceptualization is broader than the distinctions considered in OntoClean, other aspects can be analyzed, and thus the approach using UFO gives more guidelines to evaluate conceptual models than OntoClean does.

## 7. FINAL CONSIDERATIONS

In this paper we presented the new advances made in reengineering a fragment of the Software Process Ontology (SPO) originally defined in [Falbo and Bertollo 2009], making further distinctions than the ones made in [Guizzardi et al. 2008]. The newest version was obtained from the conceptual alignment of the concepts and relations defined in the SPO with the concepts and relations of the Unified Foundational Ontology (UFO). The focus was to address the conceptualization related to software processes at the light of software project planning.

The use of UFO proved to be useful for identifying problems and for driving the ontology reengineering, especially expliciting the ontological commitments that differs between standard process, project process, scheduled process and process occurrence. It became clear that a standard process is a plan described by a normative description (plan description) recognized by the organization. A project process is an internal commitment of the organization; while a scheduled process is an appointment, which refers to a time interval. Finally, process occurrences are complex actions. These distinctions also apply to the activity level. Moreover, we treated human resource allocation.

The new version of the SPO is now being used as the basis for integrating software tools for supporting software project planning. Three of these tools were developed in the context of the Ontology-based software Development Environment – ODE [Falbo et al. 2003]: DefPro, ControlPro and AlocaODE. DefPro supports defining standard software processes and tailoring them to define project processes; AlocaODE deals with allocating people to perform the project activities; and ControlPro supports project tracking, showing the project process and the state of its activities, as well as allowing people to register the hours spent in performing them. Since these tools are developed in the context of the ODE Project, they share the same conceptual model. Endeavour Agile ALM (http://endeavour-mgmt.sourceforge.net/), on the other hand, is an open source tool supporting project management and tracking, among others. We are interested in integrating Endeavour to ODE in order to add facilities for supporting scheduling by means of project Gantt charts.

Endeavour and ODE has several commonalities, but their conceptual models are very different. Our

approach is to use the new version of the SPO as an interlingua to align the concepts and relations from their conceptual models. To illustrate how this alignment is being done, take the case of project activities. In ODE, there is a class Activity that has an attribute status. This class is related to the class Activity Execution, in the following way: an Activity has none or one (0..1) Activity Execution. An Activity Execution, in turn, is related to exactly one (1..1) Activity. Activity Execution has two attributes (startDate and endDate) that represent the period of the activity execution. In Endeavour, there is a class Task with several attributes, among them status, startDate and endDate.

Both ODE and Endeavour do not explicitly distinguish from project activities (as intentions), scheduled activities (as appointments), and activity occurrences (as actions), as done in SPO (see Figure 15). In ODE, an Activity is a Project Activity in the sense of SPO when it does not have an Activity Execution associated to it. It is a Scheduled Activity when it has an Activity Execution associated to it, and its status is "Inactive" or "Awaiting Authorization". Finally, it is an Activity Occurrence when it has an Activity Execution associated to it, and its status is "In Execution" or "Completed". In Endeavour, a Task is a Project Activity in the sense of SPO when its start and end dates are null. It is a Scheduled Activity when dates are set to these attributes and the task status is "Pending". Finally, a task is an Activity Occurrence when its status is "In Progress" or "Completed". These mappings are being used to integrate the tools.

## REFERENCES


Barcellos, M. P. and Falbo, R. A. Using a foundational ontology for reengineering a software enterprise ontology. In *The Joint International Workshop on Metamodels, Ontologies, Semantic Technologies and Information Systems for the Semantic Web*. Lecture Notes in Computer Science 5833. Gramado, Brazil, pp. 179–188, 2009.

Barcellos, M. P., Falbo, R. A., and Dal Moro, R. A well-founded software measurement ontology. In *Proc. of Int'l Conf. on Formal Ontology in Information Systems*. Toronto, Canada, pp. 213–226, 2010.

Falbo, R.A. Natali, A., Mian, P., Bertollo, G., and Ruy, F. ODE: Ontology-based software development environment. In *Proc. Congr. Argentino de Ciencias de la Computación*. Argentina, pp. 1124–1135, 2003.

Falbo, R. and Nardi, J. Evolving a software requirements ontology. In *Proceeding of Conferencia Latinoamericana de Informática*. Santa Fe, Argentina, pp. 300–309, 2008.

Falbo, R. A. and Bertollo, G. A software process ontology as a common vocabulary about software processes. *International Journal of Business Process Integration and Management* vol. 4, pp. 239–250, 2009.

Gruber, T. R. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies,* 43 (5-6): 907–928, December, 1995.

Guarino, N. and Welty, C. Evaluating ontological decisions with ontoclean. *Communications of the ACM* 45 (2): 61–65, 2002.

Guizzardi, G. *Ontological foundations for structural conceptual models*. Ph.D. thesis, Centre for Telematics and Information Technology, Enschede, 2005.

Guizzardi, G., Baião, F. A., Lopes, M., and Falbo, R. A. The role of foundational ontologies for domain ontology engineering: An industrial case study in the domain of oil and gas exploration and production. *International Journal of Information System Modeling and Design* 1 (2): 1–22, 2010.

Guizzardi, G., Falbo, R. A., and Guizzardi, R. S. S. Grounding software domain ontologies in the unified foundational ontology (UFO): The case of the ODE software process ontology. In *Proceedings of the Iberoamerican Workshop on Requirements Engineering and Software Environments,*. pp. 244–251, 2008.

ISO. ISO 10006: Quality management systems – guidelines for quality management in projects, 2nd edition, 2003.

ISO/IEC. ISO/IEC 12207: Systems and software engineering – system life cycle processes, 2nd edition, 2008.

Liao, L., Qu, Y., and Leung, H. K. N. A software process ontology and its application. In *Proceedings of the First International Workshop on Semantic Web Enabled Software Engineering*, 2005.

PMI. *A Guide to the Project Management Body of Knowledge*. PMBOK guide. Project Management Institute, 2008.

Silva, E. O., Lisboa Filho, J., and Gonçalves, G. Improving analysis patterns in the geographic domain using ontological meta-properties. In *Proc. of Int'l Conf. on Enterprise Information Systems*. pp. 256–261, 2008.

SEI. CMMI for development, version 1.3, technical report CMU/SEI-2010-TR-033. , November, 2010.

Welty, C., Mahindru, R., and Chu-Carroll, J. Evaluating ontology cleaning. In *Proceedings of the National Conference on Artifical Intelligence*. pp. 311–316, 2004.