

Towards an Ontology Pattern Language for Harmonizing Software Process related ISO Standards

Fabiano B. Ruy^{1,2}, Ricardo A. Falbo¹, Monalessa P. Barcellos¹ and Giancarlo Guizzardi¹

¹ Ontology and Conceptual Modeling Research Group (NEMO), Computer Science Department
Federal University of Espírito Santo, Vitória, Brazil
+55-27-4009-2167

² Informatics Department, Federal Institute of Espírito Santo, Campus Serra, Serra, Brazil
{fabianoruy, falbo, monalessa, gguizzardi}@inf.ufes.br

ABSTRACT

Many efforts have been made for modeling and standardizing software processes. ISO/IEC JTC1/SC7, the ISO sub-committee responsible for software and systems engineering, is one of the most important groups devoted to this task. However, standards developed by this committee are frequently inconsistent and even contradictory. This led to the need for an ISO Study Group to investigate the creation of an ontological infrastructure to establish a common conceptualization for underpinning all SC7 standards. This ISO initiative is a work in progress, which has focused on the software process domain and, in particular, considering the ISO/IEC 24744 standard. In this paper, we advocate in favor of using an Ontology Pattern Language (OPL) as the main component of this ontological infrastructure. We present ISP-OPL (ISO-based Software Process OPL), an OPL that can be applied as a basis for harmonizing software process-related standards, favoring reuse when building aligned specific software process ontologies for SE sub-domains. In order to illustrate its application, we also present an ontology about the Requirements Engineering process, developed by using ISP-OPL.

Categories and Subject Descriptors

- Computing methodologies ~ Ontology engineering
- Software and its engineering ~ Software development process
- Software and its engineering ~ Patterns

General Terms

Design, Standardization, Languages.

Keywords

Ontology patterns, ontology pattern language, semantic interoperability, standards harmonization, software process.

1. INTRODUCTION

A permanent challenge in Software Engineering (SE) is to deal with quality aspects, improving the resulting products with higher productivity and lower costs. Since the quality of a software

product depends heavily on the quality of the software process used to develop it, software organizations are investing more and more in improving their software processes. In this context, several process-related quality standards and maturity models, such as ISO/IEC 12207 [8], ISO/IEC 15504 [10], and CMMI [18], are used to guide software organizations efforts towards quality software processes. These initiatives attempt software process improvement by means of disseminating best practices in an organized and standardized way. However, most of the models and standards are created independently, without necessarily sharing the same semantics. This frequently gives rise to inconsistencies between them. This problem is amplified when different standards are used together, causing semantic interoperability problems [16]. Even standards proposed by the same standardization organization have this problem. This is due to the fact that each standard defines its own scope, structure of process entities, terms and definitions, amongst other things [13].

The International Organization for Standardization (ISO) recognizes this problem. The SE standards developed by ISO/IEC JTC1's SC7 (the ISO sub-committee responsible for software and systems engineering standards) frequently employ terms whose definitions vary significantly across standards. In order to treat this problem, ISO created, in 2012, a study group to develop an ontological infrastructure aiming to be a single coherent underpinning for all the SC7 standards [7]. The goal is to establish a basic set of definitional ontologies, which can be used to derive more specific ontologies. These specific ontologies are meant to address different SE sub-domains (e.g., Software Testing), which in turn are the subject of specific SC7 standards (e.g., ISO/IEC 29119) [7]. The ISO initiative is a work in progress, which has focused on the definitional ontologies, taking mainly ISO/IEC 24744 [11] into account. The goal is to develop a Definitional Elements Ontology (DEO) and an aligned Configured Definitional Ontology (CDO) based on ISO/IEC 24744, which could be extended for building Standard Domain Ontologies (SDOs).

We argue that this basic set of definitional ontologies (DEO and CDO) should be represented as core ontologies on software processes, from which the more specific SDOs could be derived. According to [17], a core ontology provides a precise definition of structural knowledge in a specific field that spans across different application domains in this field. Moreover, we argue that, by following a pattern-oriented approach, a core ontology can systematically become more modular and extensible [3].

A core ontology for the ISO harmonization initiative should be:
(i) flexible enough for allowing ontology engineers to explore

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

SAC'15, April 13-17, 2015, Salamanca, Spain.

Copyright is held by the authors. Publication rights licensed to ACM.

ACM 978-1-4503-3196-8/15/04...\$15.00

<http://dx.doi.org/10.1145/2695664.2695796>

alternative models in the design of specific ontologies for the various software process sub-domains; (ii) modular, in order to allow the ontology engineer to select the ontology fragments relevant to the problem at hands and then reuse it; and (iii) broad enough to cover the general concepts in the software process universe of discourse. For achieving these characteristics, we argue that this core ontology should be organized as an Ontology Pattern Language (OPL). An OPL is a network of interconnected domain-related ontology patterns that provides support for solving a class of ontology development problems for a specific domain. An OPL offers a set of interrelated domain patterns, and a process with explicit guidance on what problems can arise in that domain, informing the order to address these problems, and suggesting one or more patterns to solve each specific problem [3].

A core ontology should be precise. This is achieved by basing the core ontology on a foundational ontology [17]. Thus, as the starting point for this work, we performed an ontological analysis of the ISO/IEC 24744 metamodel [16] in the light of the Unified Foundational Ontology (UFO) [5]. Based on the results obtained from this ontological analysis, and inspired on the first version of a Software Process OPL presented in [3], we define here the first version of the ISO-based Software Process OPL (ISP-OPL).

The main purpose of ISP-OPL is to provide a sound solution for the derivation of ontologies in the ISO initiative. This first version of ISP-OPL focuses on the project (or endeavor) level, and addresses three main aspects dealt by ISO software process standards: Work Units, including patterns to define the decomposition, dependence, and scheduling of work units; Work Products, considering the nature of software process work products and how they are handled; and Human Resources, dealing with how people are organized in teams, allocated to tasks and perform work units. To evaluate the applicability of ISP-OPL, we developed an ontology for the Requirements Engineering Process considering ISO/IEC 12207, ISO/IEC 29148, and other ISO related standards.

This paper is organized as follows. Section 2 discusses the ISO standard harmonization initiative and the notion of Ontology Pattern Language. Section 3 presents ISP-OPL. Section 4 illustrates how ISP-OPL can be applied in the development of more specific process ontologies, considering the Requirements Engineering process. Section 5 discusses related work. Finally, Section 6 presents our final considerations.

2. ISO STANDARD HARMONIZATION AND ONTOLOGY PATTERN LANGUAGES

Standard harmonization is very important for organizations that seek to solve multiple needs at their different hierarchical levels by using multiple standards [13]. In these cases, standards are frequently used in combination. For instance, organizations use general standards for system development, along with standards that expand on specific processes such as software testing or risk management [7]. Moreover, frequently, organizations also want to combine standards from different sources [6, 13].

Harmonious combination of standards is aided when the standards use consistent concepts. At the beginning of 2012, in the ISO SC7 plenary meeting, a set of problems was raised, among them the following [7]: (i) there is no guidance on how to build a new standard ensuring that it is compatible with other SC7 standards;

(ii) clashes in the terminology and in the semantics are observed in the current standards. Resulting from the discussion in this meeting, a study group was created, charged with the goal of investigating the potential utility of ontologies for rationalizing SC7's suite of SE International Standards [7].

This study group has proposed a layered framework comprising an ontology network [7]. In the top of the proposed framework, there is the Definitional Elements Ontology (DEO), which provides definitions for concepts, and constraints that dictate how they must be related. From DEO, a Configured Definitional Ontology (CDO) can be defined. The only CDO being worked to date is a CDO for the ISO/IEC 24744 metamodel (Software Engineering Metamodel for Development Methodologies - SEMDM) [11]. From a CDO, ontologies specific to particular standards, called Standard Domain Ontologies (SDOs), can be derived. The framework also considers in the future, to extend DEO by considering ontological distinctions put forward by foundational ontologies [5]. This extension is called Advanced Foundational Ontology for Standards (AFOS) [7].

SEMDM is the main basis for the entire framework, providing semantics for all ISO/SC7 standards. However, for the success of such initiative, the consistency of this ontological basis is crucial. Thus, in [16], we performed an ontological analysis of SEMDM in the light of the Unified Foundational Ontology (UFO) [5]. With this approach, we aim at providing a truly ontological foundation to the ISO framework. Moreover, we do not need a new foundational ontology (AFOS), but we can rely on an existing foundational ontology, in this case UFO [5]. In [16] we identified several consistency problems in SEMDM fragments, and reengineered these model fragments, based on our ontological analysis.

The CDO based on the SEMDM is meant to be reused and extended in the development of several SDOs for specific software processes, such as Requirements Engineering process (ISO/IEC 29148) and Software Testing process (ISO/IEC 29119). For this reason, ontology patterns (OPs) arise as a promising alternative to organize the ontology framework, maintaining the actual benefits, and improving it to a modular and reusable solution [3]. In such approach, a domain ontology typically results from the composition of several OPs, with appropriate dependencies between them, plus the necessary extensions based on specific needs [1]. However, in order to truly favor reuse, organizing OPs in catalogues is not enough. A pattern language can provide a stronger sense of connection between the patterns, since it expresses several types of relationships among them, such as relations of dependence, temporal precedence of application, or mutual exclusion [3].

An Ontology Pattern Language (OPL) aims to provide holistic support for using domain-related OPs in ontology development. To ensure a stable and sound application of patterns, the patterns are presented in a suggested application order. OPLs encourage the application of one pattern at a time, following the order prescribed by paths chosen throughout the language [3].

In the next section, we present the ISO-based Software Process OPL (ISP-OPL), which has been developed aiming at supporting the ISO Harmonization Initiative.

3. An OPL for ISO Software Processes

The aspects addressed by the current version of ISP-OPL are: Work Units, Human Resources and Work Products. The patterns in ISP-OPL were extracted from the reengineered fragments resulting from the ontological analysis of the SEMDM [16], as well as from the Enterprise OPL (E-OPL) proposed in [4].

Figure 1 presents a UML activity diagram showing the language paths of the current version of the ISP-OPL. As suggested in [3], in this activity diagram, Domain-Related Ontology Patterns (DROPs) are represented by action nodes (the labeled rounded rectangles); initial nodes (solid circles) represent entry points in the OPL, i.e., DROPs in the language that can be used without solving other problems first; control flows (arrowed lines) represent the admissible sequences in which DROPs can be used; merge points (diamond-shaped symbols) represent the merge of paths in the OPL; join/fork nodes (line segments) represent the conjunction of paths (join) or independent and possibly parallel paths (fork); finally, an extension to the original UML notation (dotted lines with arrows) is used to represent variant patterns, i.e. patterns that can be used to solve the same problem in different ways. Moreover, patterns are grouped according to the software process aspect to which they are related (Work Units, Human Resources and Work Products).

As Figure 1 shows, ISP-OPL has three entry points. The ontology engineer should choose one of them, depending on the scope of the specific software process ontology being developed. The

modeler can choose EP1, when the requirements for the new ontology include the definition and planning of work units; EP2, if the ontology should consider only the execution of work units; and EP3, to model only the structure of work products.

Through entry point EP1, the ontology engineer needs to choose one of (or both) the patterns WUC and WUD to model work unit definition structure. These patterns are used to represent work units defined in an endeavor, without planning a time frame for them. WUC (Work Unit Composition) represents the mereological decomposition of work units, specializing Work Unit into Process, Composite Task and Simple Task. WUD (Work Unit Dependence) deals with the dependence between work units. The pattern PPD (Project Process Definition) captures the link between a Process and the Project to which it is defined. The WUS (Work Unit Scheduling) pattern should be used to represent the time frame of a scheduled work unit, defining its planned start and end dates. Next, the ontology engineer can focus on modeling performed work units, i.e. work units already executed. Performed work units, as past events, have actual start and end dates. The tracking of performed work units against defined work units is treated by PWUT (Performed Work Unit Tracking), which relates a Scheduled Work Unit to a Performed Work Unit caused by the former. The group encompassing the patterns PWUC (Performed Work Unit Composition) and PWUD (Performed Work Unit Dependence) uses a structure similar to WUC and WUD. Additionally, the Project in which a Process is performed can be modeled with the pattern PPP (Project Process Performing).

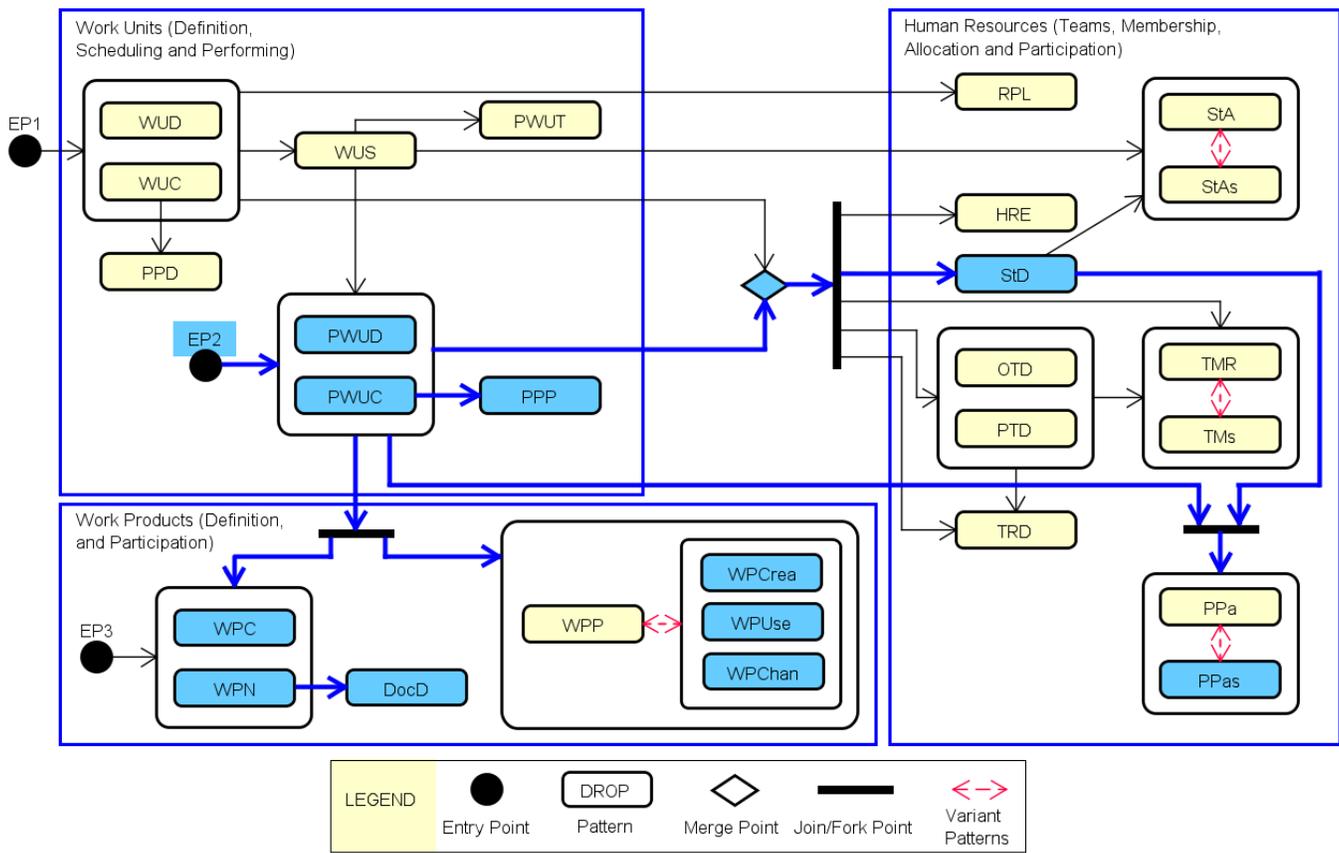


Figure 1. ISO-based Software Process OPL – Language

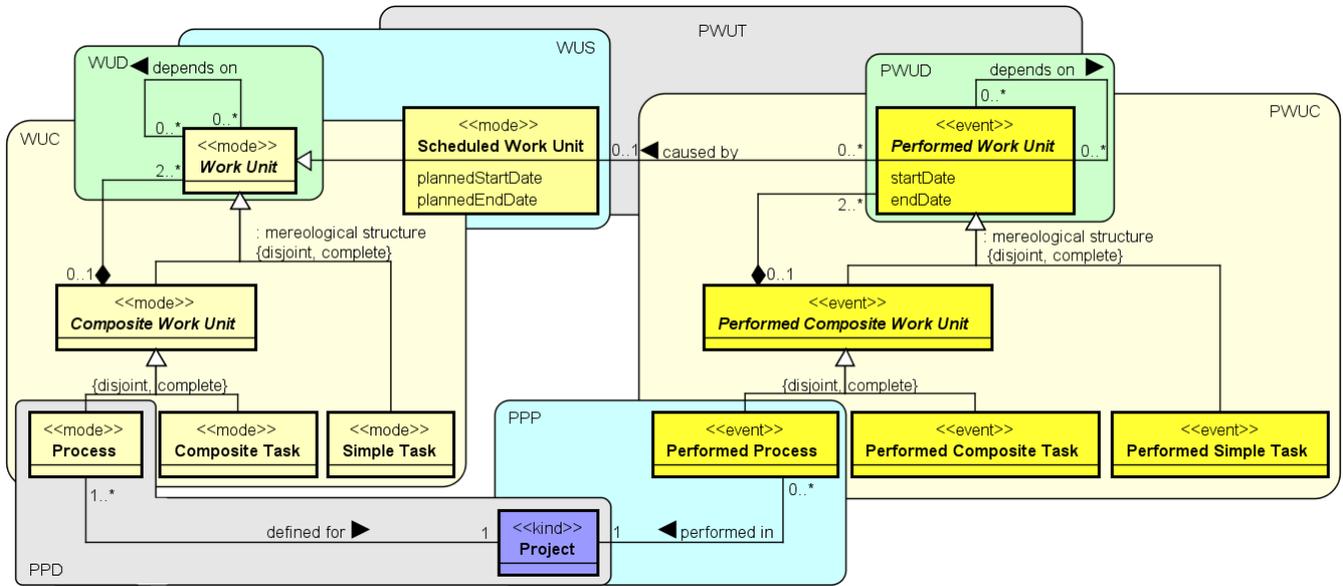


Figure 2. Patterns of the Work Unit Group

If the requirements for the ontology involve only performed work units, the entry point is EP2, allowing using only the patterns PWUC, PWUD and PPP. Figure 2 shows the complete model of the Work Unit group of patterns, detaching each one of its patterns. Every pattern (of Figure 2 and the following) is represented in the OntoUML notation. OntoUML is a UML profile that enables making finer-grained modeling distinctions between different types of classes and relations according to the ontological distinctions put forth by the Unified Foundational Ontology (UFO) [5].

After modeling Work Unit related aspects, the ontology engineer can address human resource problems by applying the patterns shown in the Human Resource group of patterns (see Figure 1). The HRE (Human Resource Employment) pattern makes the relation between an Organization and a Person, which assumes the Human Resource role. This pattern was adapted from the Employment pattern from the Enterprise OPL (E-OPL) [4]. The StD (Stakeholder Definition) pattern defines the concept of Stakeholder (someone involved in the Project), and distinguishes between two types of stakeholders: Person Stakeholder and Team Stakeholder. The OTD (Organizational Team Definition) and PTD (Project Team Definition) patterns are used to define organizational and project teams, respectively. Both are also adapted from homonymous patterns from the E-OPL. Additionally, the RPL (Role Planning) pattern can be used to model the roles responsible for performing a defined work unit, while the TRD (Team Role Definition) pattern can be applied to represent the roles a team can play. The modeler can also choose one of the alternative patterns TMs (Team Membership) and TMR (Team Membership with Role) to represent the membership relation between a team and its members (persons).

To represent the allocation of stakeholders to a scheduled work unit, there are two alternative patterns: StA (Stakeholder Allocation) and StAs (Stakeholder Allocation simplified). StA models the relational property (relator) Stakeholder Allocation that glues the stakeholder to the scheduled work unit and the organizational role the stakeholder plays. Moreover, the planned

start and end dates for the stakeholder allocation are captured. StAs is a simplified version that omits the relator, capturing only the material relation linking stakeholders to scheduled work units.

Finally, for dealing with the participation of stakeholders in performed work units, the ontology engineer can choose between the alternative patterns PPa (Producer Participation) and PPas (Producer Participation simplified). The difference between these patterns is related to whether the relator Producer Participation is explicitly represented or not (respectively). Figure 3 shows some patterns of the Human Resource group, focusing on the definition of stakeholders and their participation in performed work units.

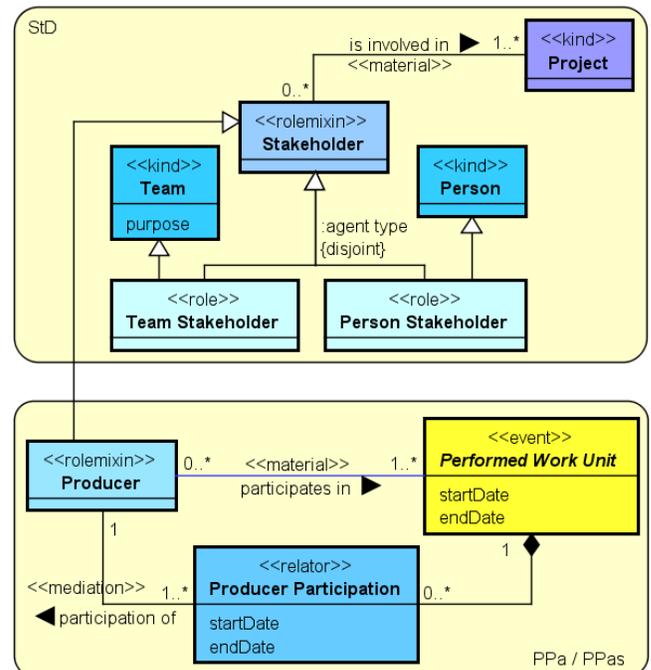


Figure 3. Human Resource Patterns (StD, PPa and PPas)

The last group of patterns constituting this OPL is the group related to Work Product patterns. This group can also be achieved through the entry point EP3, which is to be chosen when the ontology engineer wants to represent only the structure of work products. The WPC (Work Product Composition) pattern allows modeling work product mereological decomposition. WPN (Work Product Nature) is related to types of work products (such as Document, Model and Information Item). Once applied WPN, DocD (Document Depiction) can be used to model the fact that documents depict other work products. When the patterns for work unit execution are already applied (through EP1 or EP2), beyond the work product structure, the ontology engineer can also model work products handling. In this case, the WPP (Work Product Participation) pattern sets the participation of work products in performed work units. The relator Work Product Participation is modeled with its specializations for creation, usage and change participation. Alternatively, these three participation types can be modeled only by means of the corresponding material relations using the patterns WPCrea, WPUse and WPChan (Work Product Creation, Usage and Change, respectively). Figure 4 presents the complete model for the Work Product group of patterns.

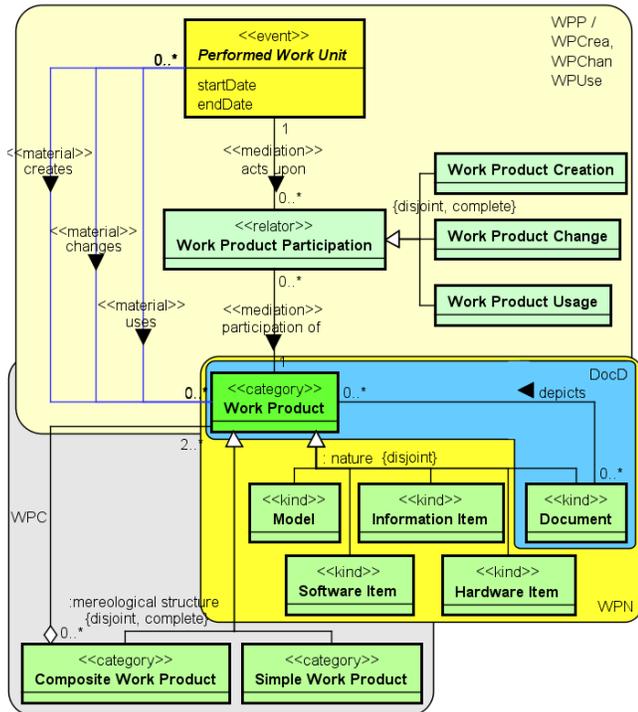


Figure 4. Patterns of the Work Product Group

It is important to highlight that, since the patterns constituting ISP-OPL are described in OntoUML, they carry out the ontological and formal semantics of this language's modeling constructs such as *kind*, *category*, *role mixin*, *relator*, *mode*, *mixin*, *material relation*, etc. OntoUML is itself a pattern-based language (albeit a domain-independent one), whose modeling primitives are patterns that embody the micro-theories comprising

the foundational ontology UFO [6]. As a consequence, the patterns of ISP-OPL are systematically constructed via the manifestation of the ontology-based patterns of OntoUML and UFO. For instance, in the patterns of Figures 2 and 4, we have the direct manifestation of the UFO pattern (micro-theory) of *Mereological Relations* [5]. Moreover, in Figure 4, we have the direct manifestation of the OntoUML *Relator* pattern [5]. Finally, in Figure 3, we have the manifestation of *Roles with Multiple Disjoint Allowed Types* pattern, or simply, the *Role Mixin* pattern [5]. As one will be able to observe in the next section, the structures constituting these patterns are carried out and presented in the ontologies created using ISP-OPL (see Figure 5).

The complete specification of ISP-OPL version 1.0, describing in detail each pattern, is available at <http://nemo.inf.ufes.br/OPL>. In the next section, we illustrate the use of ISP-OPL by building a Requirements Engineering Process Ontology. Patterns in darker rounded rectangles and thicker line paths in Figure 1 show how we have used ISP-OPL for developing this ontology.

4. REQUIREMENTS PROCESS ONTOLOGY: ISP-OPL APPLICATION

Software processes encompass a wide number of domains, such as Requirements, Architecture, Design, Project Management, Quality Assurance, Measurement, Risks, etc. For several of them there are standards covering its definitions, activities and related assets. In the context of the ISO Harmonization Initiative, beyond of the core knowledge about software process (aimed to be represented by the definitional ontologies), it is necessary to represent each of these covered domains. Moreover, it is important that the domain models may be derived from CDOs, originating each of the required SDOs. This section demonstrates how this derivation process can be supported by the application of ISP-OPL.

The domain we have chose here is the Requirements Engineering (RE) process, due to its importance as a basis for software development and its definition in several ISO standards. The RE Process Ontology is derived from ISP-OPL according to the information extracted from selected ISO SC7 standards, namely: ISO/IEC 15288:2008 – System life cycle processes [9], ISO/IEC 12207:2008 – Software life cycle processes [8], and ISO/IEC/IEEE 29148:2011 – Requirements Engineering [12]. These are the main standards dealing with requirements processes, from which our competency questions were defined. Together, the standards define three requirements-related processes: *Stakeholder Requirements Definition*, *System Requirements Analysis*, and *Software Requirements Analysis*. Due to space limitation, here we present only the sub-ontology addressing the first process, *Stakeholder Requirements Definition* (Section 6.4.1 in ISO 12207 and ISO 15288; and Section 6.2 in ISO 29148). Figure 5 shows the conceptual model of this sub-ontology. On the top, the concepts with colored background are the ones defined as part of the ISP-OPL patterns. On the bottom, the concepts with blank background are the specific ones from the RE Process Ontology. Relations in the RE Process Ontology are specializations of the homonymous relations in the OPL. Cardinalities were omitted for the sake of legibility.

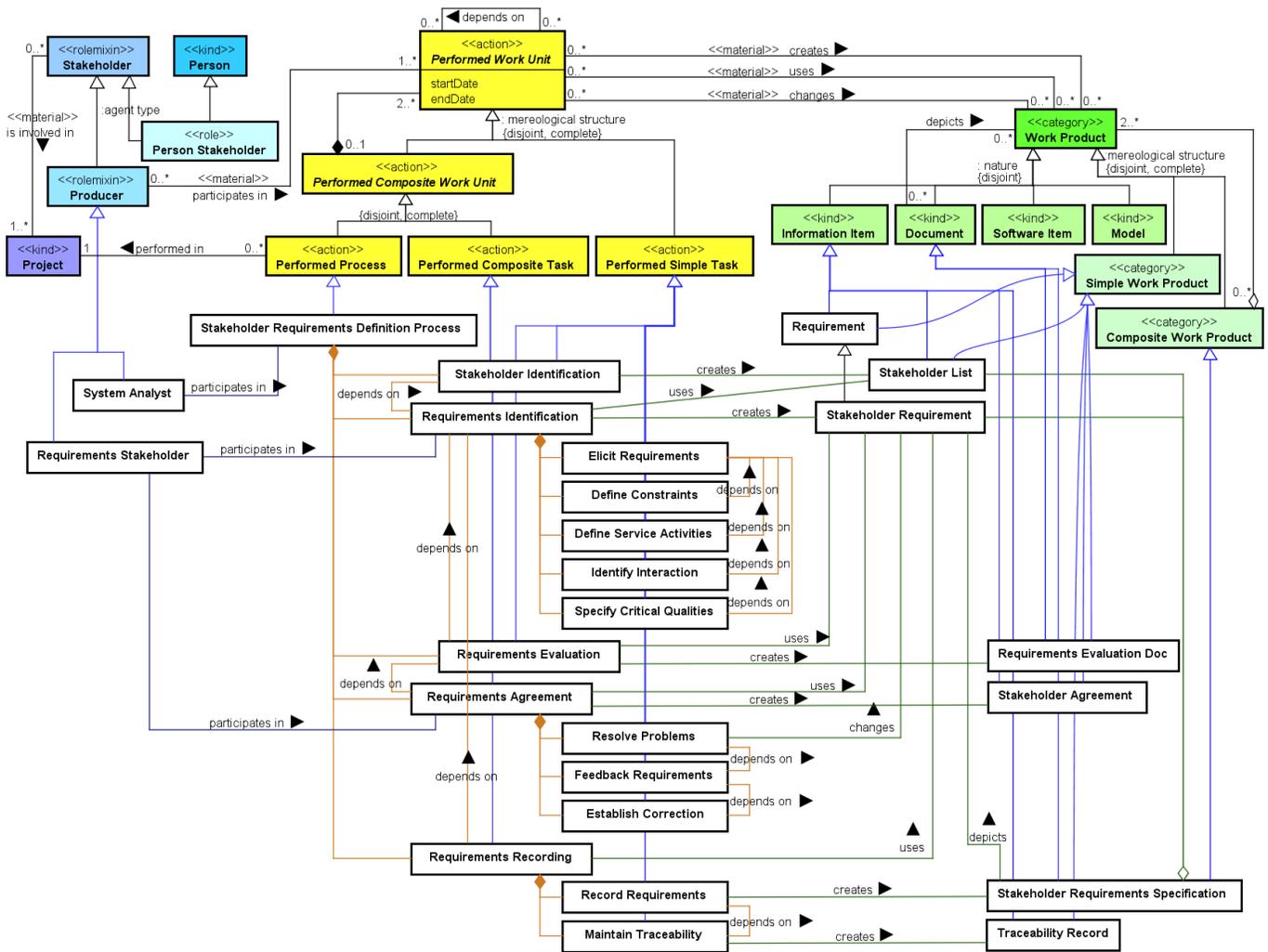


Figure 5. The Requirements Process Ontology (Stakeholder Requirements Definition Process sub-ontology)

We are interested in describing the execution of requirements processes, including the participations of human resources and work products, as it is the case of organizations adopting these standards in their projects. Thus, we start using ISP-OPL through the entry point EP2. As defined in the aforementioned standards, the *Stakeholder Requirements Definition* process is decomposed in activities, which, in turn, are decomposed in tasks. Thus, we start with the pattern PWUC, modeling the decomposition of performed work units. The *Stakeholder Requirements Definition Process* is a subtype of **Performed Process**. This specialized process is composed of five work units: *Stakeholder Identification*, *Requirements Identification*, *Requirements Evaluation*, *Requirements Agreement* and *Requirements Recording*. The first and fourth work units are **Performed Simple Tasks**, and the others are **Performed Composite Tasks**, decomposed into simple tasks as shown in Figure 5.

Another pattern considered useful here is PWUD, which defines dependencies between work units. Although the selected standards do not explicitly set dependencies between tasks, some of them can be easily inferred from the nature of work units and work products handled, as well as by considering the RE literature. Thus, we applied PWUD and established dependencies

between the work units, as shown in Figure 5. Still regarding work units, the last pattern applied is PPP, establishing the connection between the **Performed Process** and the **Project** wherein it is performed.

Once the work units are addressed, we can represent human resources. Due to the general nature of the standards, few information is given about human resources participating in work units. Thus, we have modeled only the stakeholder definition and its relation with work units. The first pattern applied is StD, in order to establish the stakeholder structure to be adopted. In the modeled process, there are only two types of stakeholders, *System Analyst* (suggested, but not explicitly named in the standards), and *Requirements Stakeholder*. Both are **Person Stakeholders** involved in the **Project**. Aiming to represent the participation of stakeholders in work units, the pattern PPA is used, specializing stakeholders as **Producers**, in order to participate in **Performed Work Units**.

The other path of ISP-OPL we followed is through the use of work products patterns. Once we have different types of work products, it is useful to distinguish between them by applying the pattern WPN. Two subtypes of **Work Product** are considered:

Information Item and **Document**. In the context of the *Stakeholder Requirements Definition Process*, we identified the following subtypes of **Information Item**: *Requirement* (in turn, specialized into *Stakeholder Requirement*), *Stakeholder List*, *Stakeholder Agreement*, and *Traceability Record*. Moreover, two types of **Document** are considered: *Requirements Evaluation Doc*, and *Stakeholder Requirements Specification* (or *StRS*, as referred by ISO 29148). The *StRS* is the main result of this process and aggregates the *Stakeholder List* and the set of *Stakeholder Requirements*. Thus, using the WPC pattern, we establish *StRS* as a **Composite Work Product** (the only one in the ontology), composed of *Stakeholder Requirements* and *Stakeholder List* (both **Simple Work Products**). Additionally, by applying the pattern DocD, *StRS*, as a document, also depicts the *Stakeholder Requirements*.

Finally, by using the patterns WPCrea, WPUse and WPChan, we established the relationships of *creation*, *usage* and *change* between the work units of the *Stakeholder Requirements Definition Process* and the corresponding work products.

The RE Process ontology, created from the application of ISP-OPL, is able to precisely define the concepts and relations for the requirements domain according to the ISO standards. These definitions, aligned to the core process definitions, serve as a common semantic basis for the related standards, contributing to their harmonization.

5. RELATED WORK

Regarding works on software process standards harmonization, Pardo and colleagues [13, 14] have developed a framework for harmonizing multiple-models using ontologies. Their concerns are the same of ours, about standards interoperability. However, whilst our work focuses on the establishment of ontologies for the domains dealt by the standards, the ontology proposed by Pardo et al. (H2mO – Ontology for the Harmonization of multiple-models) focuses on the harmonization domain itself. The main goal of H2mO is the assignment of a formal and clear definition of the most widely used techniques, methods and related terms in harmonization of multiple models [13]. It copes with concepts such as *Harmonization*, *Integration* and *Comparison* to represent the mappings between models. Although it contemplates more specific concepts such as *Process*, *Activity* and *Resource*, they are used only to map the information acquired from the models. Another important difference is about the application focus. H2mO is used for harmonizing different models applied by an organization. The ontology is used to perform comparison operations (intersection, union, difference and complement) between models, resulting in information about the related models, which helps their integrated adoption by organizations. The focus of ISP-OPL is to promote harmonization on the standards level. The main idea is to represent the knowledge about the software process domain in a reusable way to create standard domain ontologies (SDOs), establishing a semantic base of harmonized concepts to guide standards creation and revision.

Concerning ontology patterns, OPL is a new concept, established in [3], and there are few works published. The first one was the Software Process OPL [3], built from a mature Software Process core ontology grounded in UFO [2]. ISP-OPL was built from the ontological analysis of the ISO/IEC 24744 metamodel [16] in the

light of UFO, packaging the resulting ontology fragments into patterns to compose the OPL. This process of patterns definition was inspired by the patterns organization of SP-OPL, given that both these languages address the same underlying domain. SP-OPL is for general use of software processes and has patterns regarding organizational standard process, software and hardware resources and procedures. In one hand, ISP-OPL has been designed to meet the ISO harmonization initiative needs. Thus, due to the initial priorities of the ISO initiative, these aspects were not included yet in ISP-OPL. On the other hand, ISP-OPL has established finer-grained patterns, and has more specialized human resource patterns. In particular, it details the composition and nature of work products, as well as its participations in work units, and applies a terminology and structure aligned to ISO SC7 standards.

Another related OPL is the one for the Enterprise domain (E-OPL) [4]. Although constructed in a domain that is different from that of ISP-OPL, the ontology reuse intents motivating E-OPL are the same. Moreover, there is an intersection between the software process and enterprise domains regarding human resources. Once we have some analogous requirements, certain E-OPL pattern solutions motivated ISP-OPL patterns. Thus, the E-OPL patterns concerning employment, team definition and human resource membership have inspired the ISP-OPL corresponding patterns, namely HRE, OTD, PTD, TRD, TMR, and TMs, which used a similar solution adapted to the new needs and terminology.

Finally, the ISO ontological framework [7] is also related to this work. The framework does not consider ontology patterns, but provides two mechanisms for ontology derivation. The first one is based on discarding ontology parts. The idea is that the elements in the definitional ontologies are interconnected and the relations between two concepts may have a minimal cardinality of zero. This means that, for any occurrence of the concept on one side of the relation, it may have no occurrence on the other side. In this case, the concept in the opposite side (and the relation) could be discarded in a derived ontology [7]. We think discarding concepts and relations is not a matter of cardinalities, but is related to the ontology scope and the domain being modeled. Thus, for example, in ISP-OPL, the pattern PPP associates a *Performed Process* to exactly one (1..1) *Project*. If the resulting ontology does not need this relation, or even the concept, the ontology engineer can choose not to use PPP. In this case, independently of the cardinality values, the relation is not established.

A second mechanism used in the ISO ontological framework is the specialization of concepts in the resulting ontology. This mechanism is used there basically in the same way that it is used in OPLs [3], except for its applicability. The difference is that the framework derivation mechanisms are dealing with a whole model, and the OPL solution treats it in a modular way, reusing each of the patterns needed, following the guidance provided by the language.

6. FINAL CONSIDERATIONS

The ISO harmonization efforts have focused on the development of a layered ontological framework, wherein the semantics described by the higher levels (DEO and CDOs) can be propagated to the other levels (SDOs) [7]. In this context, ontology patterns are a promising approach, since they favor reuse of encoded experiences and good practices [15]. Additionally,

Ontology Pattern Languages (OPLs) have the potential to amplify the benefits of ontology patterns, by providing guidance through the ontology derivation process [3].

Our main goal is to equip the ISO framework with the features of OPLs, guaranteeing an ontologically consistent and standard-adherent basis that can be used to derive interoperable ontologies for ISO standards in a rich reuse process. In order to pursue this goal, we have developed an initial version of ISP-OPL, the ISO-based Software Process OPL. This OPL is based on ISO recognized software process standards, such as ISO/IEC 24744 and ISO/IEC 12207, and is grounded in the Unified Foundational Ontology (UFO).

We expect that ISP-OPL can be applied for modeling the several software process domains related to ISO SC7. As a proof of concept of ISP-OPL's usefulness, we have developed a Requirements Engineering Process ontology, using information of selected ISO standards. We could observe also in this experience, recurrent practical benefits of the use of OPLs [3, 4]. In particular, we have experience that the guidance provided by the patterns language in the process of producing domain ontologies resulted in an increased productivity in the development process, and a reduction of inconsistency problems in the produced models. As a difficulty in the development of the particular ontology put forth here, we have the difficulty in matching the information from different standards: These standards at times suffered from lack of information, but also from containing subjective and imprecise information. Some of these issues have helped us to improve ISP-OPL, others can be used to improve the standards themselves, as inputs for the harmonization efforts.

As an ongoing future work, we intend to enlarge ISP-OPL by adding new patterns. Our next steps include working on patterns to deal with techniques, software and hardware resources, and the planning of work products. We are starting to conduct additional cases of applying ISP-OPL for other relevant software standardized domains. Examples include Human Resource Management, Risks, Measurement, Maintenance, Configuration Management, Architectural Design, Documentation, and Quality Assurance. With the new patterns and with the feedback of these additional applications, we expect to enlarge and improve ISP-OPL, so it can be accepted as an effective solution for the ISO Harmonization Initiative.

7. ACKNOWLEDGMENTS

This research is funded by the Brazilian Research Agencies CAPES, CNPq (Process Numbers 485368/2013-7 and 461777/2014-2) and FAPES (Process Number 64618358/2013).

8. REFERENCES

- [1] Blomqvist, E., Gangemi, A., Presutti, V. Experiments on pattern-based ontology design. *Proc. Fifth International Conference on Knowledge Capture*, ACM, 2009, 41-48.
- [2] Bringuente, A.C., Falbo, R.A., Guizzardi, G. Using a foundational ontology for reengineering a software process ontology. *Journal of Information and Data Management*, 2, 2011, 511-526.
- [3] Falbo, R. A., Barcellos, M.P., Nardi, J.C., Guizzardi, G. Organizing ontology design patterns as ontology pattern languages. *Proc. 10th Extended Semantic Web Conference (ESWC'13)*, Montpellier, France, 2013.
- [4] Falbo, R.A., Ruy, F. B., Guizzardi, G., Barcellos, M. P., Almeida, J. P. A. Towards an enterprise ontology pattern language. *Proc. 29th Annual ACM Symposium on Applied Computing*, Gyeongju, Korea, 2014, 323-330.
- [5] Guizzardi, G. *Ontological Foundations for Structural Conceptual Models*, Universal Press, The Netherlands, 2005.
- [6] Guizzardi, G. Ontological patterns, anti-patterns and pattern languages for next-generation conceptual modeling. *33rd International Conference on Conceptual Modeling (ER 2014)*, Atlanta, USA, 2014.
- [7] Henderson-Sellers, B., Gonzalez-Perez, C., McBride, T., Low, G. An ontology for ISO software engineering standards: 1) Creating the infrastructure. *Computer Standards & Interfaces*, 36, 3, 2014, 563-576.
- [8] ISO/IEC, ISO/IEC 12207. Systems and Software Engineering - Software Life Cycle Processes, 2008.
- [9] ISO/IEC, ISO/IEC 15288. Systems and Software Engineering—System Life Cycle Processes, 2008.
- [10] ISO/IEC, ISO/IEC 15504. Information Technology - Process Assessment. Part 1: Concepts and Vocabulary, 2004.
- [11] ISO/IEC, ISO/IEC 24744. Software Engineering – Metamodel for Development Methodologies, 2007.
- [12] ISO/IEC/IEEE, ISO/IEC/IEEE 29148. Systems and software engineering - Life cycle processes - Requirements engineering, 2011.
- [13] Pardo, C.; Pino, F.J.; García, F.; Piattini, M.; Baldassarre, M.T. An ontology for harmonization of multiple standards and models. *Computer Standards & Interfaces*, 34, 2012, 48-59.
- [14] Pardo, C., Pino, F. J., Garcia, F., Baldassarre, M. T., & Piattini, M. From chaos to the systematic harmonization of multiple reference models: A harmonization framework applied in two case studies. *Journal of Systems and Software*, 86, 1, 2013, 125-143.
- [15] Presutti, V., Daga, E., Gangemi, A., Blomqvist, E. eXtreme design with content ontology design patterns. *Proc. Workshop on Ontology Patterns*, Washington, EUA, 2009.
- [16] Ruy, F.B., Falbo, R.A., Barcellos, M.P., Guizzardi, G. An Ontological Analysis of the ISO/IEC 24744 Metamodel. *Proc. 8th International Conference on Formal Ontology in Information Systems (FOIS'14)*, Rio de Janeiro, Brazil, 2014.
- [17] Scherp, A., Saathoff, C., Franz, T., Staab, S. Designing core ontologies. *Applied Ontology*, 6, 3, IOS Press, 2011, 177-221.
- [18] SEI, *CMMI for Development*, Version 1.3. CMU/SEI-2010-TR-033. Software Engineering Institute, Carnegie Mellon University, 2010.