UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

DEPARTAMENTO DE INFORMÁTICA

DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

FABIANO BORGES RUY

SOFTWARE ENGINEERING STANDARDS HARMONIZATION: AN ONTOLOGY-BASED APPROACH

DOCTORAL THESIS

VITÓRIA – ES, BRAZIL

OCTOBER 2017

FABIANO BORGES RUY

SOFTWARE ENGINEERING STANDARDS HARMONIZATION: AN ONTOLOGY-BASED APPROACH

Tese de doutorado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo (PPGI-UFES) como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação, sob orientação do Prof. Dr. Ricardo de Almeida Falbo e coorientação da Profª. Drª. Monalessa Perini Barcellos.

VITÓRIA – ES

OUTUBRO, 2017

Ficha catalográfica disponibilizada pelo Sistema Integrado de
Bibliotecas - SIBI/UFES e elaborada pelo autor

Ruy, Fabiano Borges, 1980-
R985s Software engineering standards harmonization: an ontology based approach / Fabiano Borges Ruy. - 2017.
228 f. : il.

Orientador: Ricardo de Almeida Falbo.
Coorientadora: Monalessa Perini Barcellos.
Tese (Doutorado em Informática) - Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Engenharia de software. 2. Modelos de capacitação e maturidade (Software). 3. Ontologia. 4. Semântica. 5. Integração de dados (Computação). I. Falbo, Ricardo de Almeida. II. Barcellos, Monalessa Perini. III. Universidade Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 004

Software Engineering Standards Harmonization: An Ontology-based Approach

Fabiano Borges Ruy

Tese submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Aprovada em 06 de outubro de 2017 por:

Prof. Dr. Ricardo de Almeida Falbo (Orientador)
UFES/ES

Prof. Dr. João Paulo Andrade Almeida (Examinador Interno)
UFES/ES

Prof. Dr. Vítor Estêvão Silva Souza (Examinador Interno)
UFES/ES

Gleison dos Santos Souza

Prof. Dr. Gleison dos Santos Souza (Examinador Externo)
UNIRIO/RJ

Profª. Drª. Andreia Malucelli (Examinador Externo)
PUC/PR

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Vitória-ES, 06 de outubro de 2017.

Dedication

To my beloved parents, José Maria and Joana.

Acknowledgements

This is one of the most important works I've done in my life. And it was built in four years of intense learning in diverse aspects. At this moment, I would like to say *thank you* to all the people who supported me, helped me, and also made my days happier in this rich period.

To my IFES colleagues for allowing me to work exclusively on this research.

To my NEMO colleagues for the worth discussions, the doubts, the ideas, and the deep UFO journeys.

To the NEMO professors for being always ready to share their knowledge and ideas.

To the professors Henderson-Sellers and McBride, who received me in UTS and asked everything I knew and didn't know about ontologies. To the professors Piattini and García, and all the Alarcos members for receiving me in UCLM and for the meaningful contributions.

To all my friends, for the support and funny moments, from those I made in Australia and Spain to those I have here for a long-time, from IFES and UFES, from the best trekking teams: Endurance and Pelicanos, and so on.

To all the people who participate in my experiments, even with all the effort required. Your feedback was essential to this work.

To the professors who evaluated this thesis, and the papers produced along this time, in special: Andreia Malucelli, Gleison Santos, João Paulo Almeida, Vítor Souza and Giancarlo Guizzardi. Your contributions and ideas made the work much better.

To my advisors, of course. Monalessa for the worth ideas and pleasant discussions. Falbo, for the long-time partnership (since 2000), being the best advisor I could have for this work, and an example for my professional life.

To my Parents and Sisters, and Kaylla for understanding me and supporting me all this time.

I'm really happy for concluding this work and for everything I've learned along this journey. You are all part of it! Thank you!

Abstract

Standards have been used as a way for disseminating best practices in Software Engineering (SE) and other areas. From the standards' developers' side (such as ISO and SEI), there is a continuous effort for capturing, organizing and describing these practices, resulting in reference models (e.g. CMMI-DEV, ISO/IEC 12207 and ISO/IEC 29110). From the users' side (such as software organizations), acknowledged best practices can be used for improving software processes aiming at benefits related to quality products, productivity and lower costs. In these contexts, it is often necessary to deal with multiple standards in combination, whether for providing more aligned reference models or for deploying practices from different sources. However, most of these standards are created independently, defining their own scope, structure and terminology, often not sharing the same semantics. This frequently gives rise to inconsistencies and divergences between them, harming the combined application. This lack of semantic alignment between the standards, and the consequential semantic interoperability problems, are the main issues to be addressed in this thesis.

Along the last two decades, researchers and standardization organizations have been attempting diverse approaches for supporting the combined use of related standards. We believe the next step towards standards harmonization involves a further exploration of the semantics underlying the standards, in a semantically rich approach. In this sense, supported by the Design Science Research paradigm, for addressing the harmonization of SE standards, we have: further understood the research context in a systematic mapping of the literature; built an SE ontology network (SEON); proposed a semantic-oriented approach for standards harmonization (Harmony); and evaluated the results in empirical studies.

SEON is our knowledge framework. It is built on a foundational ontology, holds the SE core notions aligned to general SE standards, and comprises a set of domain networked ontologies providing the base knowledge to be used in SE integration and harmonization initiatives. Harmony, in turn, provides an ontology-driven process for conducting standards harmonization efforts, offering practical orientations and encouraging the exploration of semantic aspects. It applies conceptual models and harmonization techniques sustained by SEON, which acts as interlingua during the harmonization. For supporting Harmony activities, we have developed Mapper, a models' mapping tool. Finally, several initiatives, including three empirical studies, have been conducted along the last three semesters, producing harmonized models for selected SE domains. These evaluations where essential for assessing and improving the proposed harmonization approach and ontology network.

Keywords: software engineering standards; standards harmonization; software process; semantic interoperability; ontologies; ontology network.

Resumo

Padrões, modelos e normas de qualidade têm sido usados como meio para disseminar boas práticas em Engenharia de Software (ES) e outras áreas. Do lado dos desenvolvedores de padrões (como a ISO e o SEI), há um contínuo esforço para capturar, organizar e descrever tais práticas, resultando em modelos de referência como o CMMI-DEV, a ISO/IEC 12207 e a ISO/IEC 29110. Do lado dos usuários (como organizações de software), práticas reconhecidas podem ser utilizadas para melhorar seus processos de software visando benefícios relacionados à qualidade de produtos, maior produtividade e redução de custos. Em ambos os casos, muitas vezes é necessário lidar com múltiplos padrões em conjunto, seja para prover modelos de referência mais alinhados ou para adotar práticas de diferentes fontes. Entretanto, a maior parte destes padrões é criada independentemente, definindo seu próprio escopo, estrutura e terminologia, e frequentemente não compartilham a mesma semântica. Isso geralmente causa inconsistências e divergências entre eles, afetando sua aplicação conjunta. A falta de alinhamento semântico entre os padrões, e consequentes problemas de interoperabilidade semântica, são as principais questões a serem tratadas nesta tese.

Ao longo das últimas duas décadas pesquisadores e organizações padronizadoras vêm tentando diversas abordagens para apoiar o uso conjunto de padrões relacionados. Este trabalho defende que o próximo passo na harmonização de padrões envolve explorar mais profundamente a sua semântica em uma abordagem semanticamente rica. Neste sentido, apoiado no paradigma de *Design Science Research*, para lidar com a harmonização de padrões de ES, foi realizado um mapeamento sistemático da literatura para melhor compreensão do contexto de pesquisa; construída uma rede de ontologias de ES (SEON); proposta uma abordagem de harmonização de padrões orientada a semântica (Harmony); e conduzidas e avaliadas iniciativas de harmonização por meio de estudos empíricos.

SEON é o arcabouço de conhecimento da abordagem. Foi construído sobre uma ontologia de fundamentação, carrega noções centrais de ES alinhadas a padrões gerais da área, e comporta um conjunto de ontologias de domínio em rede, provendo o conhecimento base a ser utilizado em iniciativas de integração e harmonização em ES. Harmony provê um processo orientado a ontologias para conduzir os esforços de harmonização de padrões, fornecendo orientações práticas e encorajando a exploração de aspectos semânticos. A abordagem aplica modelos conceituais e técnicas de harmonização, sustentados por SEON, que age como interlíngua durante a harmonização. Para suportar as atividades de Harmony, foi desenvolvida Mapper, uma ferramenta de mapeamento de modelos. Finalmente, algumas iniciativas, incluindo três estudos empíricos, foram conduzidas ao longo dos últimos semestres deste trabalho, produzindo modelos harmonizados para determinados domínios de ES. Tais avaliações foram essenciais para analisar e melhorar a rede de ontologias e abordagem de harmonização propostas.

Keywords: modelos e normas de qualidade; engenharia de software; harmonização de padrões; processos de software; interoperabilidade semântica; ontologias; redes de ontologias.

List of Figures

List of Tables

List of Acronyms

SE	Software Engineering
OE	Ontology Engineering
ON	Ontology Network
SPI	Software Process Improvement
SEON	Software Engineering Ontology Network
UFO	Unified Foundational Ontology
SPO	Software Process core Ontology
EO	Enterprise core Ontology
COM	Core Ontology on Measurements
QAPO	Quality Assurance Process Ontology
ROoST	Reference Ontology on Software Testing
RSMO	Reference Software Measurement Ontology
FOP	Foundational Ontology Pattern
DROP	Domain-Related Ontology Pattern
OPL	Ontology Pattern Language
SP-OPL	Software Process Ontology Pattern Language
SSM	Standard’s Structural Model
SCM	Standard’s Content Model
ISM	Integrated Structural Model
ICM	Integrated Content Model

Contents

1 Introduction

This chapter presents an overview of this thesis and defines the basis for subsequent chapters. It discusses the context and motivation for the proposed theme, the research hypothesis, the objectives we have focused on, the methodological aspects that have guided the work, and, finally, presents the structure of this thesis.

1.1 Context and Motivation

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 heavily depends on the quality of the process used to develop it (FUGGETTA, 2000), software organizations are more and more investing in improving their software processes (PINO et al., 2008) (JENERS et al., 2013) (CMMI INSTITUTE, 2016). In this context, several process-related quality standards and maturity models, such as CMMI-DEV (SEI, 2010), ISO/IEC 12207 (ISO/IEC, 2008) and ISO/IEC 29110 (ISO/IEC, 2011), are used to guide organizations' efforts towards quality software processes. Standards[1] provide organizations with agreed and well recognized practices, which assist them to interoperate and to work using engineering methods, reinforcing Software Engineering as an *engineering* discipline, instead of a *craft* (GARCÍA et al., 2006).

These Software Process Improvement (SPI) initiatives seek for disseminating and institutionalizing the best practices provided by the standards, making software processes more organized, standardized and predictable. Regardless of the standard applied, its implementation demands experience and knowledge in the domain and in the specific standard, and a high degree of efforts and investments to be successful (AAEN, 2003). Frequently, an organization adopts more than one standard for improving its processes and achieving market needs. The adoption of multiple standards allows an organization to exploit synergies between them. On the one hand, organizations can address coordination between different and common areas; on the other hand, the weaknesses of a single standard can be overcome by the strengths of others (JENERS et al., 2013). However, implementing a process adherent to multiple standards is not a simple task. Besides the known difficulties on implementing each standard, there is a considerable effort on harmonizing the portions from different standards dealing with the same aspects. It should be done by taking the best practices of each one and eliminating possible redundancies and conflicts for resulting in a quality software process.

[1] Henceforth, the different terms used for addressing a prescriptive document, such as Standard, Quality Model, Reference Model and Maturity Model, are referred as *Standard*.

Nowadays, a high number of standards are available. The International Organization for Standardization (ISO) alone has more than 20,000 standards, with more than 170 devoted to Software and Systems Engineering. In this context, most of the standards are created independently, by different standardization groups or organizations, defining their own scope, structure, abstraction level and terminology (BIFFL et al., 2006), not necessarily sharing the same semantics. Therefore, even the standards, which have as premise supporting the standardization of diverse process-related elements, when applied together, show inconsistencies and divergences between them (SMITH, 2006) (PARDO et al., 2012) (MCBRIDE et al., 2012).

Along the last years, some actions have been taken to mitigate this interoperability problem. Thus, standardization organizations have included in some SE standards, as complementary information, alignments / mappings to other standards. This is the case of ISO/IEC 12207 (with high-level alignments to ISO/IEC 15288 (ISO/IEC, 2008b)), ISO/IEC 29110 (with alignments to ISO/IEC 12207), and the Brazilian MR-MPS-SW (SOFTEX, 2016) (with alignments to CMMI-DEV and ISO/IEC 29110). Additionally, researchers have been working in diverse proposals for comparing, mapping, integrating and harmonizing related standards (PAULK, 1993) (ROUT, 1998) (LEPASAAR; MÄKINEN, 2002) (FERCHICHI et al., 2008) (PARDO et al., 2013) (JENERS et al., 2013) (HENDERSON-SELLERS et al., 2014) (GONZALEZ-PEREZ et al., 2016). Even standardization organizations, such as the Software Engineering Institute (SEI) and the International Organization for Standardization (ISO), recognize the problem. SEI conducts the Process Improvement in Multimodel Environments (PrIME) project (SEI, 2010b); and ISO has created a study group aiming at harmonizing its own standards (HENDERSON-SELLERS et al., 2014), the ones under ISO/IEC JTC1's SC7 (the ISO Sub-Committee responsible for Software and Systems Engineering standards).

Despite the recent advances, SE Standards Harmonization is still an open research issue and a need for standards' developers (standardization organizations) (HENDERSON-SELLERS, 2012) and users (software organizations, consultancy and assessment groups) (PARDO et al., 2013). Multimodel environments are becoming a widely common situation for software organizations (LARRUCEA; SANTAMARIA, 2014) and the interoperability issues around them must be addressed.

Standards Harmonization focuses on the combined use of techniques for dealing with multiple standards that should interoperate. It is essentially a semantic interoperability problem, as we can observe very interrelated information described in different and sometimes conflicting ways by distinct sources (standards). Dealing with this information in combination is a complex problem, and we believe its semantic nature demands a semantically-oriented solution. Firstly, we need a robust knowledge framework[2] serving as reference to deal with standards' information. In addition, we need an approach able to conduct harmonization efforts towards a harmonized view of the standards.

In particular, this thesis focuses on the problem of Harmonizing SE Standards, precisely Software Process Standards. They are the standards most used by software organizations, for conducting SPI initiatives, context in which the problem becomes more evident and has motivated diverse research efforts (PARDO et al., 2015).

We believe the next step towards standards harmonization involves a further exploration of the semantics underlying the standards, in a semantically rich approach. For harmonizing multiple standards (frequently with different backgrounds), they should be analyzed in the light of a knowledge reference, able to provide consistent representation of the domain(s) in focus. Ontologies have been used in this quest (HENDERSON-SELLERS et al., 2014) (PARDO et al., 2015). They can offer the domain knowledge in a precise and meaningful way supporting diverse decisions in harmonization efforts. Moreover, SE is a wide and complex domain. An ontological representation of this domain should be modular (for addressing different harmonization initiatives) at the same time preserving the connections between the main notions. An Ontology Network (SUÁREZ-FIGUEROA et al., 2012) can provide such organization. Thus, we consider that SE Ontologies, organized in an Ontology Network, are able to figure as the stated SE reference knowledge framework. Additionally, an ontology-based approach for harmonizing structural and content models of the subject standards can fulfill the methodological needs.

1.2 Research Hypothesis

Considering, as previously mentioned, that:

- Standards are developed by different standardization groups / organizations, and semantic alignment has not been a matter of high priority;
- Software organizations frequently adopt standards for Software Process Improvement (SPI), often in combination, and spend a high-degree of investments and efforts for deploying them;
- Researchers and standardization organizations have been attempting diverse approaches for comparing, mapping, integrating and harmonizing related standards along the last years;
- Beyond the structure, terminology or language issues, standards harmonization is a semantic interoperability problem.

[2] We mean by *knowledge framework* a body of knowledge on the subject area (in this case, SE), able to provide a well-founded and agreed representation of the involved domains, including their key notions and how they are related and defined.

The research hypothesis of this thesis is:

Software Engineering Standards can be harmonized by a systematic approach applying a grounded Software Engineering Ontology Network for semantically analyzing, comparing and integrating them.

1.3 Objectives

The general objective of this thesis is to **develop an ontology-based approach for SE standards harmonization.** This general objective is broken down into the following specific objectives:

SO1. Perform a systematic literature mapping for analyzing related studies. A mapping study provides a comprehensive view of the problem and solutions applied, thus we can better understand the topic and establish the focus of our research efforts.

SO2. Establish a SE reference knowledge framework by designing and developing a well-founded Software Engineering Ontology Network (SEON). At the heart of SEON, there shall be a core ontology on software processes. This core ontology must be aligned to general SE standards, and organized in a flexible and extensible way, allowing to increase the network domain knowledge with SE domain ontologies. Finally, the network has to be equipped with mechanisms for supporting the access, creation, integration and evolution of networked ontologies.

SO3. Develop a SE standards harmonization approach grounded in SEON. The approach must provide a process to guide harmonization endeavors exploring semantic aspects, as well as guidelines on how to perform each process step. Moreover, the approach must be assessed on its effectiveness to harmonize SE standards, and the feedback from its application shall be used to evolve both the approach and the ontology network.

1.4 Research Method

The problem of Harmonizing SE Standards to provide a unified representation of the subject standards characterizes the context of this research. The research method adopted in this work follows the Design Science Research paradigm (HEVNER et al., 2004). According to Hevner (2007), Design Science Research (DSR) is an embodiment of three closely related cycles of activities: the *Relevance Cycle*, the *Design Cycle* and the *Rigor Cycle*, as shown in Figure 1.1. *Relevance* is mainly related to: (i) research motivation, which arises from business needs and/or possible improvement opportunities in current theories, as well as (ii) “good” articulation between the proposed solution and the motivation, to reinforce the contributions. *Rigor* is associated with the use of a reliable body of knowledge (e.g., theories, methods, models, experiences, and expertise) in the research effort. Finally, *Design* concerns the core activities of the research process towards achieving the research objectives and supporting the research hypothesis. As such, *design* takes *relevance* and *rigor* aspects into account.

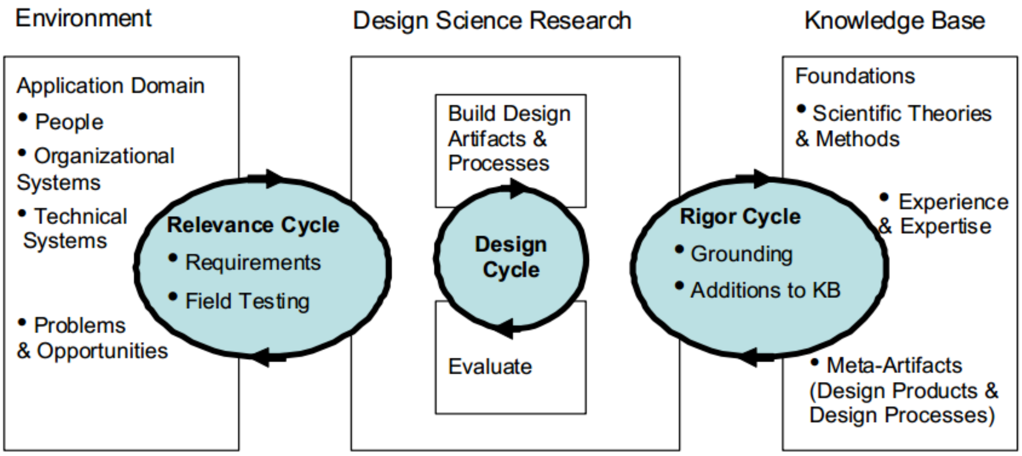


Figure 1.1. Design Science Research Cycles (HEVNER, 2007).

The ***Relevance Cycle*** initiates a design science research with an application context and defines the problem to be addressed, the research requirements and the criteria for evaluating the research results (HEVNER, 2007). The problem addressed by this work essentially focuses on the lack of semantic alignment between related SE standards. This research opportunity was identified in the literature and confirmed in the systematic mapping we have conducted. The mapping shows a panorama of the topic that allowed us to better understand the problem, existing solutions, research gaps and trends. As we could notice, although the solution proposals are evolving, in fact, SE Standards Harmonization is still an open research issue. Moreover, the doctorate candidate experienced in practice some of the pointed out difficulties while working in SPI initiatives, often involving multiple standards, as a process analyst (from 2005 to 2008) and as a consultant (from 2009 to 2013). Both advisors also got involved in SPI initiatives. Adding up to the mentioned, our group was invited in 2013 to contribute to the ISO Study Group's harmonization initiative, due to our expertise in ontologies and software processes. This sum of factors configured the main motivation for this research. Although initially involved in the ISO initiative, we aim at a wider solution, applicable for SE standards from diverse standardization organizations, and able to contribute to the standards harmonization research topic. Additionally, we believe that harmonization requires as support a knowledge framework on the subject domain, in this case, SE. Thus, this research is also an opportunity to establish an ontological knowledge framework for SE, a goal that has been motivating many efforts in our group in the last two decades (since the initial version of the software process ontology proposed in (FALBO, 1998)). This knowledge framework is SEON and it can be useful for diverse harmonization / integration initiatives, including standards harmonization, semantic documentation, and tool integration. Lastly, this research offers, in the process of building SEON, a good opportunity to improve some of our ontology engineering methods (especially those dealing with ontology patterns). These motivation factors were used for shaping this research scope, as well for inspiring the evaluation criteria for the research results. Empirical evaluations are also included in this work, aiming at assessing the applicability and effectiveness of the achieved results and to collect feedback for improvements.

The ***Rigor Cycle*** refers to knowledge use and generation. A knowledge base composed of foundations and methodologies is used to ground the research, and the knowledge generated by the research contributes to knowledge base growing (HEVNER, 2007). Rigor is achieved by appropriately applying the existing foundations and methodologies (HEVNER et al., 2004). In this work, the main foundations come from knowledge related to ontologies, SE / software process literature and standards, and studies on SE standards harmonization. The Unified Foundational Ontology – UFO (GUIZZARDI, 2005) contributes to this research by grounding most of the results we have achieved and inspiring the application of conceptual models in several activities. To cite some, UFO is used in this work: (i) for conducting an ontological analysis on a general SE standard, ISO/IEC 24744 (RUY et al., 2014); (ii) as the foundational layer of SEON, providing the general grounding for the whole network (RUY et al., 2016); (iii) as source of foundational ontology patterns (FOPs), supporting the creation of domain-related ontology patterns (DROPs) (RUY et al., 2017) and enriching the SEON's ontology derivation mechanisms; (iv) for supporting diverse decisions, as well as providing conceptual distinctions and constraints for the harmonization approach. The studies identified and analyzed in the systematic mapping also contribute by presenting the research status, techniques applied, solutions proposed, research gaps and trends, supporting relevant decisions in this research. These studies revealed a starting point and several ideas we could reuse and improve while building our harmonization approach. Moreover, we have been supported by the literature on Ontology Engineering (OE), especially in Ontology Networks and Ontology Patterns; and on SE, mainly collecting domain-specific and standards' knowledge for building SEON.

The ***Design Cycle*** concerns the development and evaluation of artifacts or theories to solve the identified problem (HEVNER, 2007). In this research, the most relevant results can be organized in two main perspectives: design artifacts and processes. Regarding the artifacts, the proposed SE knowledge framework is **SEON** (RUY et al., 2016), and it includes an ontology network architectural definition, and a grounded and well-structured base of SE ontologies. Additionally, software tools were developed to support the SEON's specification building and the approach's mapping activities. From the process's perspective, the intended harmonization approach, named **Harmony**, embraces a process based on ontologies and conceptual models for standards harmonization. Also in this perspective there are the SEON mechanisms for building and integrating domain ontologies. The artifacts and processes have evolved along this research, supported by empirical evaluations and discussions with researchers. These evaluations have been conducted for applying the solutions proposed in practical problems and collecting feedback for improvements. Firstly, we have assessed an ontology pattern language (ISP-OPL) (RUY et al., 2015b), which has contributed to the domain ontology derivation mechanisms of SEON, and shown some directions for the research. Three other empirical studies sum to this research by evaluating (and helping to improve) evolving versions of Harmony: a preliminary version for producing harmonized processes (textually) from SE standards; an intermediary version resulting in mapped and harmonized standards' models; and the last one, also creating a harmonized standards' model, but involving an expert (since the others were conducted mostly with graduate students). The empirical studies' results and feedbacks provided relevant information for evolving SEON and Harmony. Other important sources for improvement in the research artifacts and processes were the discussions performed with the NEMO professors and colleagues, and in two internships done along the research period: firstly at the University of Technology of Sydney (UTS), with professor Henderson-Sellers, leader of the ISO harmonization initiative (HENDERSON-SELLERS, et al., 2014); and then at the University of Castilla-La Mancha (UCLM), with professors García and Piattini, co-authors of the HFramework harmonization proposal (PARDO et al., 2013). Presentations of the work in progress and valuable discussions brought many insights to the present research.

In sum, the *design* reflects the main cycle that articulates the other two cycles, *rigor* and *relevance*, towards conducting the research project. Figure 1.2 summarizes the discussion about the *design*, *rigor*, and *relevance cycles*, and highlights the main elements of each cycle in the context of this thesis.

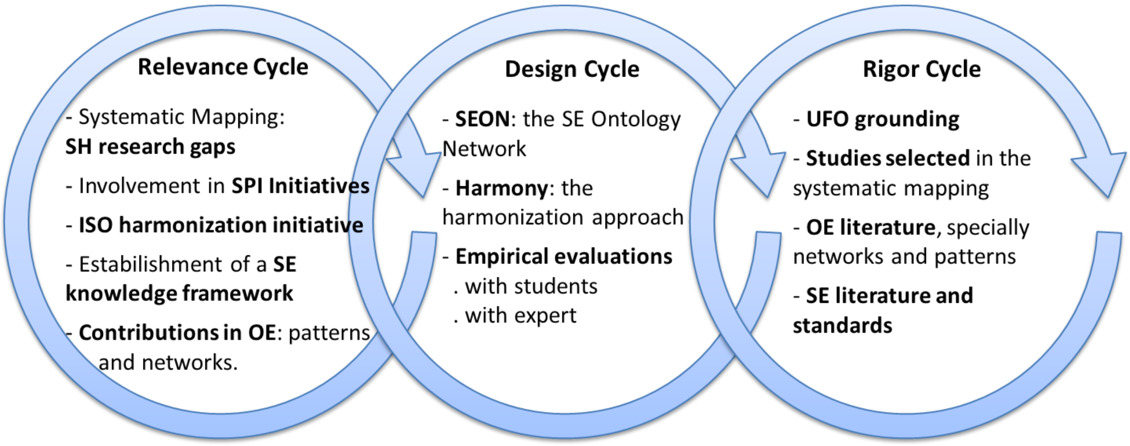


Figure 1.2. Overview of the Design Science Cycles in this research (based on (HEVNER, 2007)).

1.5 Organization of this Thesis

This chapter presents the Introduction of this thesis, involving the general aspects, namely: the context and motivation for this research, the research hypothesis and objectives, and the methodological aspects. The next chapters are organized as follows:

- **Chapter 2. Background: Standards Harmonization and Ontologies:** presents an overview of the state of the art required for grounding the ideas of this research. The content includes a discussion on the standards harmonization problem and concepts, and on the most relevant ontological notions applied along the work, including an introduction to UFO and ontological analysis, core ontologies and ontology patterns, and ontology networks.
- **Chapter 3. A Systematic Mapping on Software Engineering Standards Harmonization:** presents the systematic mapping conducted to classify and analyze related studies, specifically the ones addressing harmonization of SE standards by means of a common model. The findings and results are presented and discussed.
- **Chapter 4. A Software Engineering Ontology Network:** presents SEON, the Software Engineering Ontology Network, and discusses: SEON requirements and architecture; how the core layer is built to support the inclusion of new domain networked ontologies and to favor semantic integration / harmonization initiatives; the current networked domain ontologies; and integration mechanisms. Moreover, this chapter also introduces the SEON specification and discusses some related work.
- **Chapter 5. An Approach for Harmonizing Software Engineering Standards:** presents Harmony, our harmonization approach, and describes its requirements, process and guidelines, as well as a proof-of-concept. Additionally, the supporting mapping tool is presented, and Harmony is contrasted against related works.
- **Chapter 6. Evaluating Harmony:** presents the empirical evaluations conducted for assessing the approach and their main results and conclusions.
- **Chapter 7. Final Considerations:** summarizes the main ideas discussed in this thesis, addressing the research contributions and the impacts of this work, the current limitations, and, finally, perspectives of future work.
- **Appendix A: Harmonizing Quality Assurance Processes:** presents the report of our proof-of-concept harmonizing the Quality Assurance processes of CMMI-DEV and ISO/IEC 12207.
- **Appendix B: Questionnaire and Interviews Scripts:** presents the questions used for the questionnaires and interviews applied in the empirical studies.
- **Annex A: Harmonizing Software Testing Processes:** presents the report of the empirical study harmonizing the Software Testing processes of ISO/IEC 12207, ISO/IEC 29110 and ISO/IEC 291119.

2 Background: Standards Harmonization and Ontologies

This chapter presents an overview of the background for this research. It introduces the standards harmonization problem and concepts; and describes the most relevant ontological notions applied along the work, including some ontology classifications, the Unified Foundational Ontology (UFO), ontological analysis, core ontologies, ontology patterns and ontology networks.

2.1 Standards Harmonization

Several standardization organizations around the world are developing standards for a variety of areas. Some of the most known are ISO (International Standardization Organization), IEC (International Electrotechnical Commission) and IEEE (Institute of Electrical and Electronics Engineers). More specifically, in the SE area, there is SEI (Software Engineering Institute) and, in Brazil, SOFTEX (Association for Promotion of Brazilian Software Excellence). These organizations are responsible for thousands of standards for diverse domains (such as Engineering, Telecommunication, Health, Agriculture and so on). ISO alone has more than 20,000 standards, being almost 3,000 for IT, and 176[3] specifically for Software and Systems Engineering (ISO/IEC, 2017). The ISO JTC1 - SC7 (Subcommittee 7: Software and Systems Engineering, in the Joint Technical Committee 1: Information Technology) deals with standards covering several aspects within this area, as Figure 2.1 shows. Regarding evaluations, ISO 9001, the most adopted standard, reached over one million certified organizations; CMMI has more than 13,000 assessments (CMMI INSTITUTE, 2016); and the MPS-SW model more than 700 evaluations, only in Brazil (SOFTEX, 2017).

[3] For reference of the growth, SC7 had 147 standards in 2013, the beginning of this research.

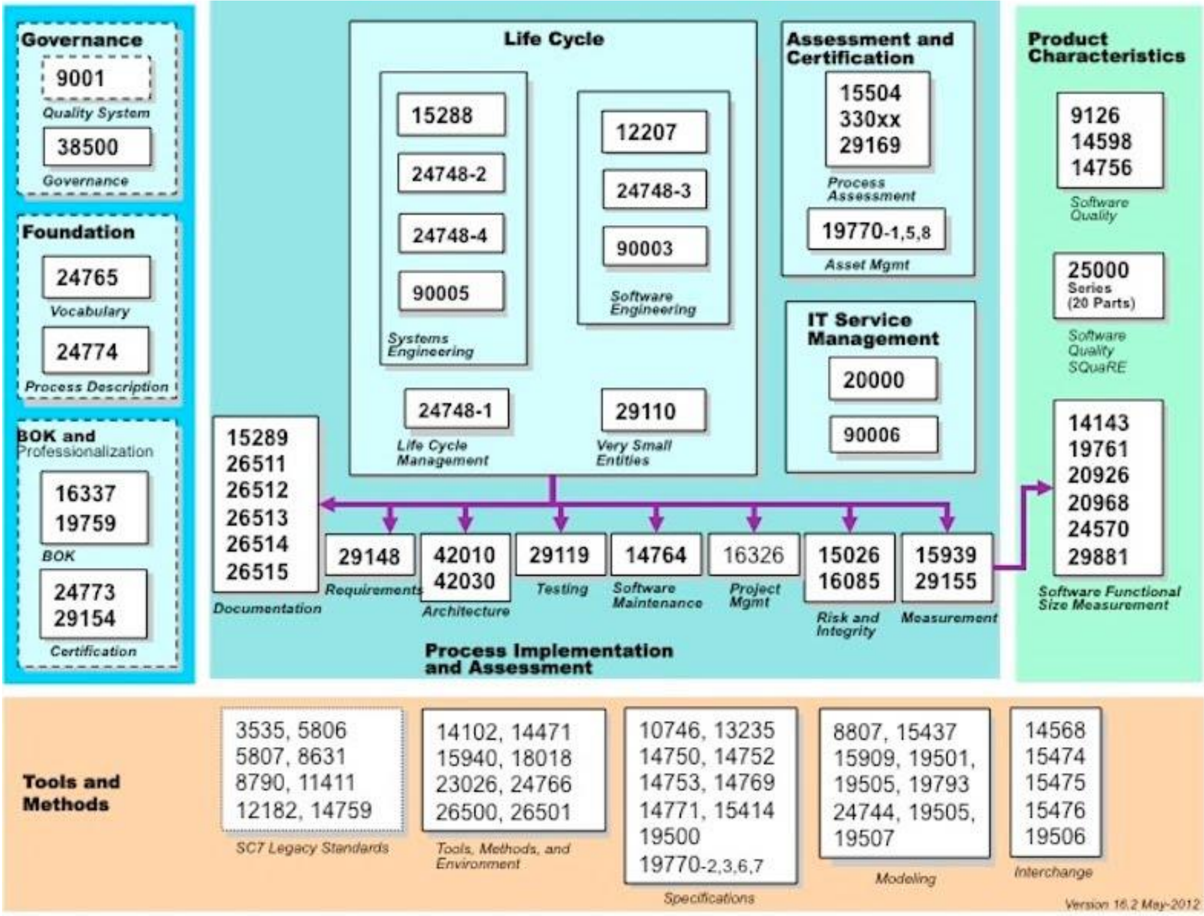


Figure 2.1. Standards published by ISO/IEC JTC1 / SC7 (ISO, 2013).

Software organizations adopt some of these and other quality standards with varied arrangements. Some use them for supporting the same or overlapping areas, taking advantage of the best provided practices; some for complementary areas, covering different organizational aspects. However, although several standards attempt an alignment with the most referenced ones, they are still built with high independence, and usually present incompatibilities when deployed in combination.

When analyzed together, it is possible to observe diverse aspects affecting the standards. Espindola (2011) points out the main problems indicated by researchers working on the standards integration issue:

- Diversity in the adopted vocabulary (LEPASAAR; MÄKINEN, 2002) (YOO et al., 2006) (FALBO; BERTOLLO, 2009) (PARDO et al., 2012);
- Diversity in the representation structures (LEPASAAR; MÄKINEN, 2002) (YOO et al., 2006) (KELEMEN et al., 2012);
- Diversity in the abstraction levels (BALDASSARRE et al., 2009) (FERREIRA et al., 2010);
- Granularity differences between elements of distinct standards (PARDO et al., 2012) (ROUT et al., 2001);

- Difficulty in tracking process assets and practices or requirements (YOO et al., 2006) (KELEMEN et al., 2012);
- Subjectivity in the identification of equivalences between the compared components (YOO et al., 2006) (BALDASSARRE et al., 2009).

These problems are mostly related to the diversity of knowledge contained in distinct standards and how it is represented / organized. The considered notions have not a common semantic background, and, thus, use to adopt multiple, sometimes conflicting, definitions. To illustrate that, in a search in the SE Vocabulary provided by ISO/IEC 24765 SEVOCAB[4], it is possible to find five or more different definitions for common concepts such as *process*, *activity*, *product* and *document*.

Moreover, in an ISO initiative for harmonizing its own standards (the SC7 ones), McBride and colleagues (2012) present a document entitled "The growing need for alignment", listing the following issues:

- Increasing recognition that standards are becoming multi-disciplinary and that there is no guidance for a new team when building a standard to ensure it is compatible with other (SC7) standards;
- Clashes of terminology and subtle clashes in semantics;
- Impacts of external legislation;
- Need to move from serendipitous knowledge of such problems to organizational (SC7-level) solutions;
- An ontology / taxonomy should be produced, in particular using a recognized terminology (e.g. ISO 704:2009 – Terminology work – principles and methods).

Due to these problems, some initiatives have been conducted for harmonizing SE standards. Standardization organizations and researchers have been working in diverse proposals for homogenizing, comparing, mapping, integrating, unifying and harmonizing related standards. These initiatives are analyzed and discussed in Chapter 3. However, before that, it is important to precisely define what each of these techniques means. We have based our definitions mostly in the classification provided by Pardo and colleagues (2012), reaching the following definitions:

- **Homogenization** is the adaptation of standards to a pre-defined structure, to put different standards in a homogeneous format (same structure), easing the application of other techniques.
- **Comparison** is the analysis of standards or their processes seeking for similarities and differences to know the level of equality of such elements.

[4] SEVOCAB does not intend to be a collection of precisely defined core concepts, but simply a collection of existing definitions, 'mined' from the definitional clauses of various standards. It is available for searches on https://pascal.computer.org/sev_display/index.action.

- **Mapping** is a lower-level comparison of the standards considering structural or content elements (such as *activities*, *work products* and *roles*).
- **Integration / Unification** is the definition of a new (integrated) schema combining selected practices or definitions of different standards.

The Harmonization notion, as our main purpose in this work, deserves a further discussion. According to (SIVIY et al., 2008), "*Harmonization <u>is not</u> about creating a master meta-standard or a new single standard that encompasses all other standards, or about declaring any single combination of standards as the best, or suggesting a universal combination to suit all. Rather, it is about developing an appropriate solution to meet your individual organizational objectives. To accomplish this, harmonization requires understanding and leveraging the properties of the technologies of interest as well as composing these properties and the process architecture into a harmonized solution*". Additionally, Pardo and colleagues (2012) state that Harmonization is *the development of one suitable solution that allows organization's goals to be satisfied*. We agree with these definitions; however, we expect to apply harmonization in a wider context, not exclusively for deploying a set of standards in an organization. We also intend to consider harmonization in the context of standardization organizations (e.g. for standards development or review) and research projects (e.g. for building integrated models or identifying similarities and divergences). Hence, the definition we follow in this work is:

- **Harmonization** is the <u>combined use of a set of techniques</u> (such as homogenization, mapping and integration) to provide a <u>solution regarding the interoperability</u> of two or more standards.

Besides the techniques' definitions, at this moment it is important to define more precisely two other concepts regarding the perspectives we use to deal with the information the standards provide. Each standard is generally organized into a (pre-)defined structure used to classify the more specific information the standard aims to provide. Thus, each standard can be observed into two different perspectives:

- **Standard's Structure**: is the set of higher-level standard's elements used to categorize / organize the standard's contents. These structural elements can be represented as a structural model of the standard, applied to different standard's portions (or sections). For example, *process*, *activity* and *task* (in ISO/IEC 12207); and *process area*, *goal* and *practice* (in CMMI-DEV) are structural elements.
- **Standard's Content**: is the specific knowledge provided by the standards, organized according to the standard's structure. These content elements, usually described as plain text, fill the standard with the specific practices / rules / information to be observed. For example, in CMMI-DEV, *Requirements Engineering* (classified as a process area), *Manage Requirements* (classified as a goal) and *Obtain an Understand of Requirements* (classified as a practice) are content elements.

The harmonization techniques can be applied to one or both perspectives. For example, Homogenization can be applied only to the standard's structure once the goal is to create a **structural alignment** between different standards. Mapping, Integration and Harmonization could be applied to both structural and content perspectives. Overall, the harmonization of standard's contents can be better conducted when standard's structure is already aligned (or previously harmonized).

Quality standards, in general, are developed by domain experts; carry out an acknowledged background knowledge; can have some shared conceptualization (when developed by the same group or inspired in other standards); but are not developed with a common conceptual ground. Thus, ontologies have been applied for achieving a sound solution in standards harmonization (PARDO et al., 2012) (HENDERSON-SELLERS et al., 2014) (RUY et al., 2014). They can provide a shared consistent conceptualization, aligned to reality, and mapped to the standards' definitions. Next sections discuss some ontological notions we have applied in this research. Furthermore, Chapter 3 presents a mapping study covering the research initiatives on SE standards harmonization using a common model.

2.2 Ontologies

The term *ontology* has been used in Philosophy since the 17th century to refer both to a philosophical discipline, and as a domain-independent system of categories that can be used in the conceptualization of domain-specific scientific theories (GUIZZARDI, 2007).

In the past decades, there has been a growing interest in the subject of ontology in computer and information sciences (GUIZZARDI, 2007). In Computer Science, we refer to an ontology as a special kind of information object or computational artifact (GUARINO et al., 2009). Gruber (1993) originally defined the notion of an ontology as an "*explicit specification of a conceptualization*". Borst (1997) adapted this definition for a "*formal specification of a shared conceptualization*". This definition additionally required that the conceptualization should express a shared view between several parties, a consensus rather than an individual view. Studer et al. (1998) merged these two definitions stating that: "*an ontology is a formal, explicit specification of a shared conceptualization*."

Ontologies have been classified in diverse perspectives in the literature, for example, according to their levels of generality, formality, applicability, etc. In this research, we are mainly interested in the classification criterion regarding *Generality Levels*, in which ontologies can be classified into *Foundational*, *Core* and *Domain ontologies* (SCHERP et al., 2011). At the highest generality level, **Foundational Ontologies** span across many fields and model the most basic and general concepts and relations that make up the world (including domain-independent notions, such as *object*, *event*, *dependence*, *classification*, *parthood relation* etc.) (GUARINO, 1998). **Domain Ontologies**, in turn, describe the conceptualization related to a specific domain (such as *Requirements* and *Testing* in

Software Engineering, or *Electrocardiogram* in *Medicine*) (GUARINO, 1998). **Core Ontologies**, located between the foundational and domain levels, provide a definition of structural knowledge in a specific field, but one that still spans across different application areas in this field (such as *Services*, *Enterprises* and *Measurement*). These ontologies are typically built based on foundational ontologies and provide a refinement to them by adding detailed concepts and relations in their specific fields (SCHERP et al., 2011).

The different generality levels do not amount to a discrete classification, but to a continuum (FALBO et al., 2013), ranging from foundational ontologies that are totally domain-independent (such as DOLCE (GANGEMI et al., 2002) and the Unified Foundational Ontology - UFO (GUIZZARDI, 2005)), to domain ontologies, for a very particular domain (such as *Software Requirements* or *Quality Assurance*). Finally, core ontologies, despite being more general than domain ontologies, are also domain-dependent.

Higher-level ontologies can be used to support the development of lower-level ontologies, i.e., foundational ontologies can be used as basis for building core and domain ontologies, and core ontologies can support the development of domain ontologies. In fact, considering the continuous nature of the aforementioned classification, some ontologies can be used for supporting the development of more specific ontologies even within the same level of generality. For example, UFO-A (an ontology of endurants) and UFO-B (an ontology of events), both of which are foundational ontologies, have been used as basis for building UFO-C (an ontology of social entities). The latter, albeit being more specific, is still considered to be a foundational ontology. SPO (a Software Process core Ontology) (BRINGUENTE et al., 2011) is grounded mainly in UFO-C, while ROoST (an ontology for the Testing domain) (SOUZA et al., 2013) is developed by extending SPO. The *Reference Software Requirements Ontology* (RSRO, presented in Chapter 4), in turn, is a domain ontology that is mostly grounded directly in UFO-C. Finally, the *Runtime Requirements Ontology* (RRO) (DUARTE et al., 2016), a domain ontology on requirements at runtime, was developed based on RSRO. Figure 2.2 illustrates the view of ontology generality levels as a continuum using the aforementioned ontologies. The dashed arrows show the grounding dependencies between the ontologies in different levels.

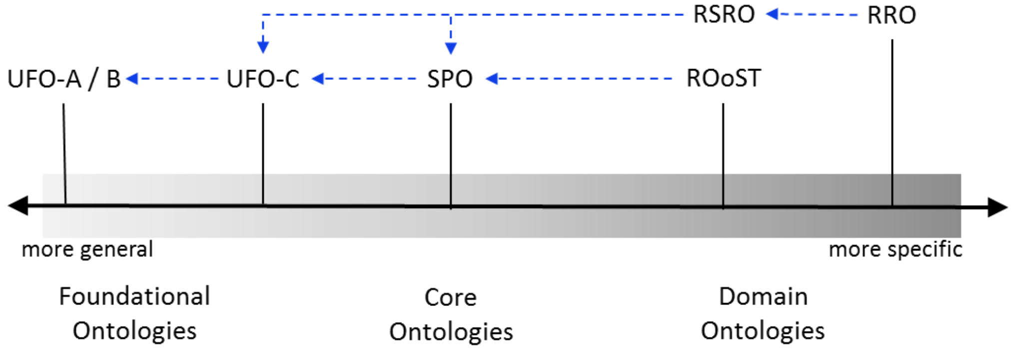


Figure 2.2. Ontology Generality Levels as a Continuum (adapted from (FALBO et al., 2013)).

Another relevant classification criterion concerns the intended application of ontologies. Guizzardi (2007) makes an important distinction between ontologies as conceptual models (termed *reference ontologies*) and ontologies as coding artifacts (*operational ontologies*). A reference domain ontology is constructed with the goal of making the best possible description of the domain in reality. It is a special kind of conceptual model, an engineering artifact with the additional requirement of representing a model of consensus within a community (GUIZZARDI, 2007). On the other hand, once users have already agreed on a common conceptualization, different operational versions of a reference ontology can be derived. Contrary to reference ontologies, operational ontologies are designed with the focus on maximizing particular non-functional requirements (e.g., certain desirable computational properties). In other words, when developing a reference ontology, the focus is on expressivity of the representation and truthfulness to the domain being represented (*domain appropriateness*), even at the expenses of computational characteristics such as tractability and decidability (GUIZZARDI, 2007). In summary, in the view employed here, a reference ontology is a particular kind of conceptual model, namely, a reference conceptual model capturing the shared consensus of a given community.

In this thesis, the focus is on addressing a semantic interoperability problem regarding standards harmonization. Even with the support of methods and tools, it is still an activity essentially performed by humans. Thus, in this text, when referring to *ontologies*, we mean *reference ontologies*. Regarding the generality levels, all of them are applied, as discussed in Chapter 4. Since in this thesis we use UFO as basis for building ontologies for harmonization purposes, next we briefly present it.

2.3 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 (GUIZZARDI, 2005). It is composed of three main parts:

- UFO-A, an ontology of endurants (objects) (GUIZZARDI, 2005);
- UFO-B, an ontology of perdurants (events) (GUIZZARDI et al., 2013); and
- UFO-C, an ontology of social entities (both endurants and perdurants) (GUIZZARDI et al., 2008), built on top of UFO-A and UFO-B.

It has been successfully employed as a basis for analyzing, reengineering and integrating many modeling languages and standards in different domains (e.g., UML, TOGAF, ArchiMate, RM-ODP, TROPOS/i*, AORML, ARIS, BPMN), as well as for the development of core and domain ontologies in different areas. Examples of targeted domains include Services, Capabilities, Organizational Structures, Communities, Goals and Motivations, Constitutional Law, Business Processes, Discrete

Event Simulation, Simulation for Land Covering and Use, Measurement, and Software Engineering, among others (GUIZZARDI et al., 2015). Of all applications of UFO, one deserves special attention, namely, the use of UFO categories and axiomatization in the design of an ontology-driven conceptual modeling language, OntoUML (GUIZZARDI, 2005). It was conceived as an ontologically well-founded version of the UML 2.0 fragment of class diagram. 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 UFO (GUIZZARDI, 2005).

Among the available foundational ontologies (such as DOLCE (GANGEMI et al., 2002) and GFO (HELLER; HERRE, 2004)), we choose UFO, because it has been constructed with the primary goal of developing foundations for conceptual modeling. Consequently, UFO addresses many essential aspects for conceptual modeling, which have not received a sufficiently detailed attention in other foundational ontologies. Examples are the notions of material relations and relational properties. For instance, this issue did not receive up to now a treatment in DOLCE, which focuses solely on intrinsic properties (qualities). Moreover, UFO has been employed in many semantic analyses (such as (ALMEIDA et al., 2010) (EESSAAR; SGIRKA, 2013) (GUIZZARDI et al., 2008)) and has already been used for grounding SE ontologies that are essential for our research, as it is the case of SPO (BRINGUENTE et al., 2011).

In the sequel, the most important distinctions of UFO used in this research are presented. This description is based mainly on (GUIZZARDI, 2005) (CARVALHO; ALMEIDA, 2016) for UFO-A, (GUIZZARDI et al., 2013) for UFO-B, and (GUIZZARDI et al., 2008) (BRINGUENTE et al., 2011) for UFO-C.

Figure 2.3 shows a fragment of UFO-A. A fundamental distinction in UFO-A is between particulars and universals. *Particulars* (or *Individuals*) 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. Particulars are instances of Universals (a first order type: 1stOT), while Universals are instances of second order types (2ndOT). *Substantials* are existentially independent particulars (e.g. a person, a car). *Moments***,** in contrast, are particulars that can only exist in other particulars (e.g. a person’s headache, the marriage of a couple), and thus they are *existentially dependent* on them. Existential dependence can also be used to differentiate intrinsic and relational moments: *Intrinsic Moments* are dependent on only one single individual (e.g., a color, a temperature), while *Relators* depend on a plurality of individuals (e.g., a marriage, an employment). Most distinctions made for particulars also 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 2.3.

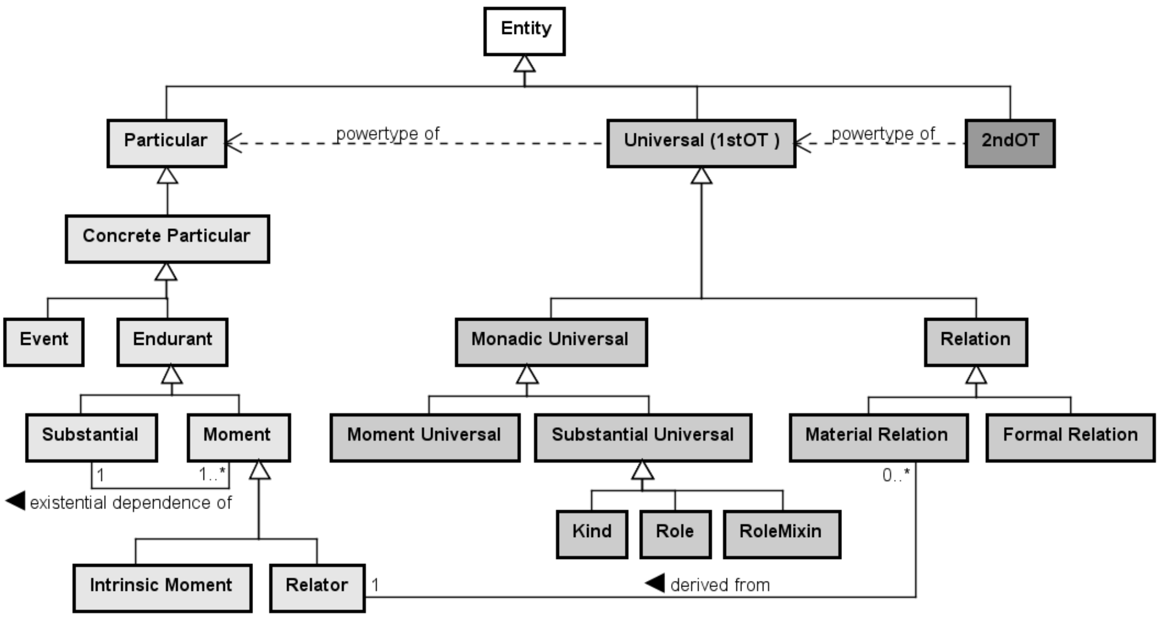


Figure 2.3. A Fragment of UFO-A – An Ontology of Endurants.

While persisting in time, *Substantial* particulars can instantiate several *Substantial Universals*. Some of these types, a substantial particular instantiates necessarily (i.e., in every possible situation) and they define what this entity is. These are the types named *Kind* (e.g., a person, an organization). There are, however, types that a substantial also instantiates in some circumstances, but not in others, such as is the case of *Roles* (e.g., student, husband). A *Role* is a type instantiated in the context of a given event participation or of a given relation. The abstractions of common properties of roles of different *Kinds* are represented by *Role Mixins* (e.g., customer, that aggregates properties of the roles individual customer and corporate customer). Both *Kind* and *Role* are *sortal substantial universals*, but *Kind* is a *rigid sortal*, while *Role* is an *anti-rigid sortal*. *Role Mixin* is an anti-rigid *mixin substantial universal*. Although not represented in Figure 2.3, *Sortal Universal*, *Rigid Sortal*, *Anti-rigid Sortal* and *Mixin Universal* are concepts of UFO-A. For details see (GUIZZARDI, 2005).

Relations are entities that link together other entities. *Formal Relations* hold between two or more entities directly, without any further intervening individual (e.g., University provides Course). *Material Relations*, conversely, have material structure of their own, deriving from a *Relator*, which mediates the related entities (e.g., Enrollment, mediating Course and Student). The relations between a relator and the connected entities are said *mediation relations*.

UFO-B makes a distinction between two types of *Particulars*: endurants and perdurants (or *Events*). *Endurants*, presented in UFO-A, are said to be wholly present whenever they are present, i.e., they *are in time*, (e.g., a person, a tree). *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 business process, a party).

A fragment of UFO-B is shown in Figure 2.4. The main category in this ontology is *Event*. They can be atomic or complex. *Atomic Events* have no proper 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 figure depicts these two aspects on which events can be analyzed, 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*.

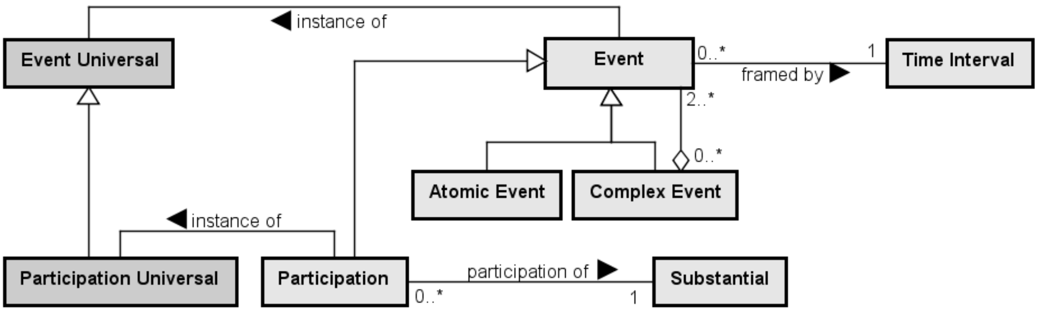


Figure 2.4. A Fragment of UFO-B – An Ontology of Events.

The third UFO's subontology is UFO-C, an ontology of social entities (both endurants and perdurants). Some fragments of UFO-C are presented in Figure 2.5 (regarding Agents and Objects), Figure 2.6 (depicting Commitments and Appointments) and Figure 2.7 (addressing Actions and Participations).

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) and social objects (e.g., money). 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 and social roles. A *Plan Description* is a special type of normative description that describes *Complex Action Universals* (plans).

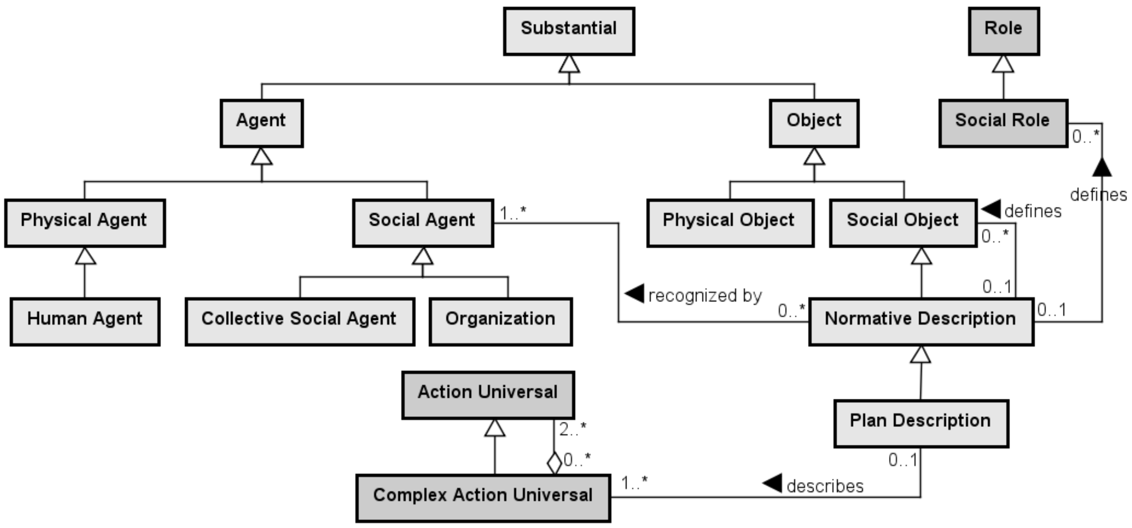


Figure 2.5. A Fragment of UFO-C: Distinction between Agents and Objects.

Agents are substantials that can bear special kinds of moments, named *Intentional Moments*. They can be *Mental* or *Social 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. Besides internal commitments (intentions), there are also social commitments. A *Social Commitment* is type of *Social Moment* establishing a commitment of an agent towards another (e.g., an agreement, a promise). A special type of *Commitment* is an *Appointment*, which is a commitment whose goal explicitly refers to a time interval (e.g., a scheduled task). Both *commitments* and *appointments* can be either *Internal* or *Social*.

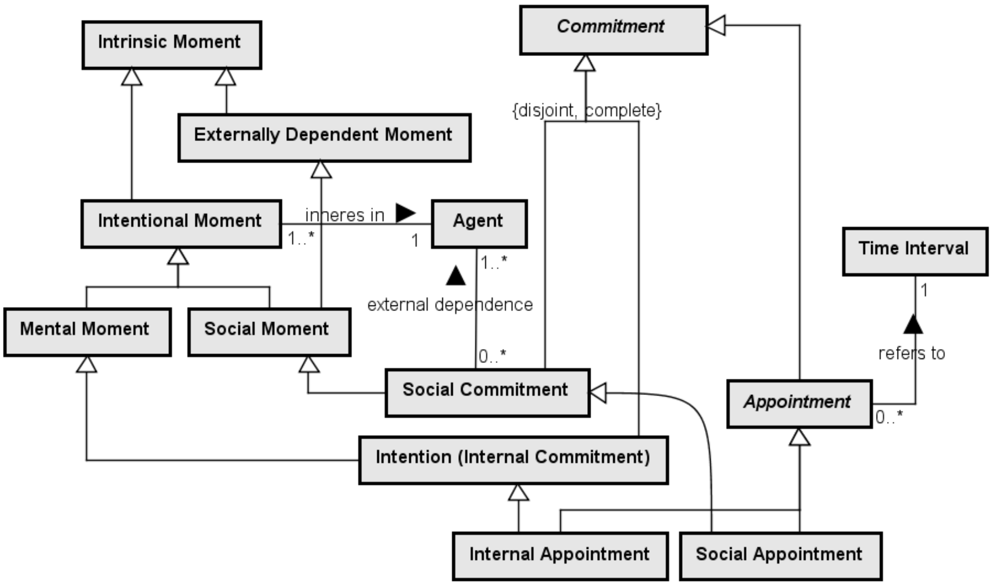


Figure 2.6. A Fragment of UFO-C: Commitments and Appointments.

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. Since actions are intentional, only agents can perform actions. An object participating in an action does not have intention. *Object Participations* can be of the following types: *Creation*, *Change*, *Usage* or *Termination*. For illustrating, a person using a hammer is a complex event with two participations: the person's participation (also an action) and the hammer usage participation.

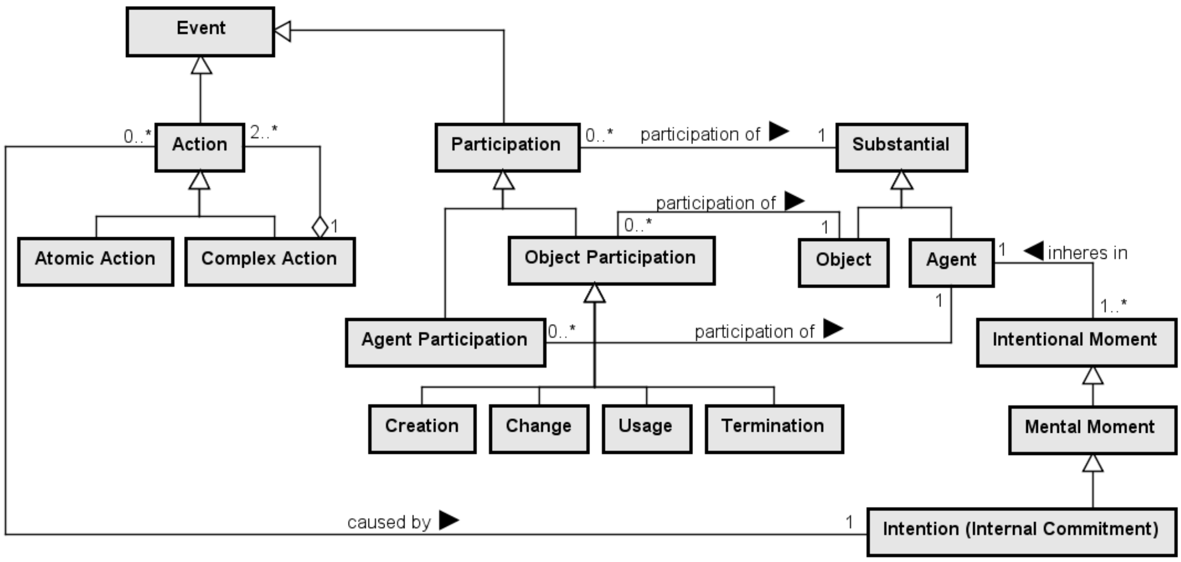


Figure 2.7. A Fragment of UFO-C: Actions and Participations.

The UFO distinctions (such as these just described) are being used in diverse applications, including providing ground for the development of core and domain ontologies and for analysing conceptual models in a foundational light. The idea behind ontological analysis is to provide a sound foundation for modeling concepts, if assumed that such concepts are aimed at representing reality (FETTKE; LOOS, 2003). Several efforts have shown the benefits of ontological analysis, such as (SHANKS et al., 2003) (SMITH, 2006) (ALMEIDA et al., 2010), which includes: (i) the rigorous definition of models, in terms of real-world semantics; (ii) the identification of problems in the definition, interpretation or usage of concepts; and (iii) recommendations for model formality improvements.

An ontological analysis process basically consists in analyzing the concepts and definitions of a subject model, evaluating if they make sense according to the chosen semantic basis. The results range from identified problems and proposed solutions to the complete reengineering of that model. The employed basis use to be a formal and well accepted semantic reference. A foundational ontology, consistently providing the basic types that describe the world, is a good choice. UFO, for example, have been applied for some ontological analysis (such as (GUIZZARDI et al., 2008) (ALMEIDA et al., 2010) (BRINGUENTE et al., 2011) (EESSAAR; SGIRKA, 2013)).

2.4 Core Ontologies and Ontology Patterns

In the second generality level, the core ontologies provide a precise definition of structural knowledge in a specific field that spans across different application areas in this field (SCHERP et al., 2011). These ontologies are typically built based on foundational ontologies and provide the domain-related core knowledge, being useful for supporting the development of more specific domain ontologies. Regarding SE, for example, a foundational ontology (such as UFO) can ground SE core ontologies (describing *Software*, *Artifacts*, *Software Processes*), which can be the basis for building domain-specific ontologies.

This is the case of SPO, a Software Process core Ontology. It was originally build in (FALBO, 1998) and evolved along the time while supporting diverse applications, including the development of a variety of SE domain ontologies. A second version of SPO was proposed contemplating evolutions in the software process area, and mapping it to software process standards (FALBO; BERTOLLO, 2009). Although these earlier versions had used a UML profile applying formal axioms, it was not yet based on a foundational ontology. Finally, UFO was used for ontologically analyzing and reengineering SPO in (GUIZZARDI et al., 2008) and (BRINGUENTE et al., 2011). From this point on, the discussed ontology building chain becomes complete, having UFO (foundational) grounding SPO (core), which, in turn, grounds several SE (domain) ontologies (as discussed in Chapter 4). Moreover, Falbo and colleagues (2013), in order to improve the support given for the development of domain ontologies, claim that core ontologies should be presented as Ontology Pattern Languages

(OPLs), and provide an SPO version organized as an OPL, SP-OPL (*Software Process Ontology Pattern Language*).

Core ontologies are conceived mainly aiming at reuse, and thus, a pattern-oriented design approach is appropriated for organizing them. Following such approach, core ontologies become more modular and extensible (SCHERP et al., 2011). Moreover, by providing a network of patterns and rules on how they can be combined, an OPL improves the potential for reusing a core ontology, by enabling the selective use of parts of the core ontology in a flexible way. This is very important due to pragmatic reasons, since ontology engineers developing specific ontologies for that domain might want to focus on selected aspects of the domain, disregarding others (FALBO et al., 2016).

An Ontology Pattern (OP) describes a particular recurring modeling problem that arises in specific ontology development contexts and presents a well-proven solution for the problem (PRESUTTI et al., 2009). Among several types of ontology patterns (FALBO et al., 2013b), we are interested in the Conceptual Ontology Patterns (COPs). COPs are modeling fragments extracted of either foundational ontologies (Foundational OPs, henceforth, FOPs) or core/domain reference ontologies (Domain-related OPs, henceforth, DROPs). They are intended for use during the ontology conceptual modeling phase, and focus only on conceptual aspects, without any concern with technological or computational issues (FALBO et al., 2013b). A COP extracted from a higher-level ontology can be used to support the development of lower-level ontologies. FOPs are reusable fragments derived from foundational ontologies (FALBO et al., 2013b), packaging a small portion of foundational structural knowledge. Since foundational ontologies are in the top-most generality level, FOPs can potentially be applied to any domain. DROPs are reusable fragments extracted from reference core/domain ontologies, packaging the core knowledge related to a domain (FALBO et al., 2013b). Thus, DROPs can be seen as fragments of a core/domain ontology that are applicable for designing ontologies of a lower generality level.

An Ontology Pattern Language (OPL) is a network of interconnected DROPs 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 (FALBO et al., 2013).

Core ontologies are valuable instruments for supporting the development of domain ontologies, especially if they are grounded in a foundational ontology (embedded with structural knowledge) and organized as an OPL (prepared for reuse). Such characteristics can be an advantageous way for enhancing the development of a series of domain ontologies in the subject area.

2.5 Ontology Networks

In the past decades, there has been a growing interest in the subject of ontology in computer and information sciences. In the last few years, this interest has expanded considerably in the context of the Semantic Web and MDA (Model-Driven Architecture) research efforts, and due to the role ontologies are perceived to play in these initiatives (GUIZZARDI, 2007). According to Suárez-Figueroa and colleagues (2012), the Semantic Web is characterized by the existence of a very large number of distributed semantic resources, which subscribe to alternative but often overlapping modeling schema (i.e., ontologies). Together, these resources define a network of ontologies related through a variety of different meta-relationships such as versioning, inclusion, inconsistency, similarity, and others. This emerging scenario is radically different from the relatively narrow contexts in which ontologies have been traditionally developed and applied, and calls for new methods and tools to support effectively the development of a new kind of network-oriented semantic applications. Hence, ontologies are not stand-alone artifacts. They relate to each other in ways that might affect their meaning, and are inherently distributed in a network of interlinked semantic resources. More precisely, a network of ontologies or an *Ontology Network* (ON) is a collection of ontologies related together via a variety of relationships, such as alignment, modularization, and dependency. Accordingly, a *networked ontology* is an ontology included in such a network, sharing relationships with a potentially large number of other ontologies (SUÁREZ-FIGUEROA et al., 2012).

Intuitively, this aspect of considering ontologies as included in a network implies that they are defined not only through their content but also in terms of ontology metadata, which provide information about their provenance, purpose, and the relations with other ontologies and semantic resources, among other things.

One of the most common ways for two ontologies to relate is to be dependent on each other. More precisely, it is often the case that, to define its model, an ontology refers to the definitions included in another ontology (SUÁREZ-FIGUEROA et al., 2012). It can be done by using an OWL primitive (*owl:imports*) in operational ontologies, or by relating or specializing a concept from other ontology in reference ontologies. It allows the ontology developer / engineer to declare such a relationship, and also adapt previous definitions to make the ontologies consistent with each other.

Aligning ontologies is a way to put different models in correspondence by declaring which entities in one ontology are in some way related to those in another ontology (or specialized / generalized or the same as). The main purpose of alignments is to ensure semantic interoperability, making it possible to merge ontologies in a meaningful way by representing information in one ontology in terms of the entities in another (SUÁREZ-FIGUEROA et al., 2012).

Large, monolithic ontologies are hard to manipulate, use, and maintain. Modular ontologies, on the contrary, divide the ontological model in self-contained, interlinked components, which can be considered independently, while at the same time participate to the definition of a specific aspect of an ontology (SUÁREZ-FIGUEROA et al., 2012).

Besides the presented characteristics, ONs embracing ontologies from the same domain can experiment a synergetic characteristic for the development of networked ontologies. As the network is being populated, more and more common entities are defined and reused, in such a way that, for adding new networked ontologies, much of the effort and definitions have already been done. This mechanism makes the development of networked ontologies more productive, and also turns the ON more consistent as a whole.

2.6 Final Considerations

This chapter presented the background on standards harmonization and ontologies for our research. We believe that SE reference ontologies can be properly organized in an Ontology Network. In addition, the networked ontologies at the higher granularity levels are suitable for being fragmented in ontology patterns and organized as OPLs, providing a valuable support for the network consistency and evolution. To conclude, we argue that our SE ontology network (SEON) is plenty applicable for supporting harmonization of SE standards. This and other aspects of the solution we are proposing are supported by the presented background. In the following, Chapter 3 presents a mapping study going deeply on the standards harmonization topic. The discussion on our proposed solution as well as its specific requirements, results and evaluations are presented in Chapters 4, 5 and 6.

3 A Systematic Mapping on Software Engineering Standards Harmonization

This chapter discusses a systematic mapping study on the research theme: Software Engineering Standards Harmonization. It presents the context and motivation for the study, the research method applied, the study results, discussions and limitations, and the final considerations towards our research goals. We also review a secondary study on the same topic that was reported in the literature by Pardo and colleagues (2010).

3.1 Introduction

Nowadays, to achieve higher levels of quality and productivity, the software industry has adopted a variety of quality models focusing on improving its processes, products, and competitiveness. We can find standards in diverse formats, domains and purposes, such as CMMI-DEV, ISO/IEC 12207, ISO/IEC 29110 and MR-MPS-SW for software development; ISO 9001 (ISO, 2015) for quality management systems; ISO/IEC 25010 (ISO/IEC, 2011b) for product quality; ITIL (ITGI, 2008), COBIT (ITGI, 2008) and ISO/IEC 20000 (ISO/IEC, 2011b) for Information Technology (IT) governance; ISO/IEC 27001 (ISO/IEC, 2013) and ISO/IEC 27002 (ISO/IEC, 2013b) for information security; and other standards for more specific domains such as measurement, requirements and testing.

These standards are often used in combination. Software organizations can benefit from selecting the best practices from different standards, but, at the same time, they need to deal with possible incompatibilities in structure, vocabulary and meaning (JENERS et al., 2013) (PARDO et al., 2013). This has taken the attention of researchers and standardization organizations. In the recent years, the Software Engineering (SE) community has demonstrated a growing interest regarding process improvement environments where multiple models are involved (PARDO et al., 2015). Additionally, standardization organizations are also concerned about providing more aligned standards, as it is the case of SEI and ISO.

In this context, diverse initiatives have been conducted to alleviate the effects of the standards incompatibilities and to present a more aligned view of them. Examples are mappings between standards (PAULK, 1993) (MUTAFELIJA; STROMBER, 2003); development of conceptual models representing knowledge relative to a single standard (SOYDAN; KOKAR, 2006) (MENDES; ABRAN, 2004) or related to a domain comprising a set of standards (GARCÍA et al., 2006) (FALBO; BERTOLLO, 2009) (BARCELLOS; FALBO, 2013); and a number of approaches and techniques aiming to homogenize, compare, map, integrate and harmonize standards (FERCHICHI et al., 2008) (JENERS et al., 2013) (PARDO et al., 2013) (HENDERSON-SELLERS et al., 2014). Since this task frequently involves knowledge representation, some of these initiatives make use of conceptual models to support the harmonization efforts and for better dealing with the standards' knowledge (JENERS; LICHTER, 2013) (RUY et al., 2015) (PARDO et al., 2015). This mapping is particularly interested in those harmonization initiatives that adopt a common conceptual model.

A mapping study provides a comprehensive overview of a research area, revealing whether there is research evidence on a particular topic. Its results help to figure out gaps in the current research status, and provide a direction to appropriately position new research activities (KITCHENHAM; CHARTERS, 2007) (KITCHENHAM et al., 2011) (PETERSEN et al., 2008). This mapping study aims at providing a broad view on the standards harmonization theme. We expect to see the evolution of the harmonization initiatives over the years, reaching the actual state of the art; and to take a better view of the problem, proposed solutions, techniques applied, standards involved, types of common conceptual models used, and how those solutions have been evaluated. In sum, we expect this mapping can support understanding the research advances thus far and establish a scenario for our research as well as future research on SE standards harmonization.

The established goal for this mapping is **to identify publications addressing harmonization of Software Engineering standards by means of a common conceptual model, and analyze the applied approaches**. Hence, the mapping scope is delimited by *publications* describing some *harmonization effort* involving *SE quality standards* and using a *common conceptual model*.

Regarding the use of common conceptual models, we are interested in papers in which a conceptual model is used as a common knowledge reference for harmonizing SE standards. Metamodels, ontologies and other types of models are often used for representing or classifying the standards' knowledge. These models can be used as an input, being a knowledge source for harmonization initiatives, or as output, being produced in the integration stages. In this mapping, we are particularly interested in those papers applying a *common conceptual model* for supporting the harmonization effort. It means that (i) the common model is necessarily used as an input for applying the harmonization techniques; and (ii) the common model must carry domain or standard general knowledge, providing in some extension a semantic reference for the harmonization.

We decided to investigate only those papers using common conceptual models as a knowledge reference for harmonization, because we see SE standards harmonization as a semantic interoperability problem. Semantic interoperability, in general, refers to the ability to exchange information based on meaning. In several contexts, semantic conflicts arise because the things being integrated (systems, data, models and so on) do not share a common conceptualization. Thus, to solve semantic conflicts, common conceptual models, such as ontologies and metamodels, can be used as an interlingua to map concepts from different sources. Semantics constitutes an important element for integration approaches. It must be correctly addressed to bring integration to their full potential. Moreover, semantic integration should occur at the knowledge level (PARK; RAM 2004), considering that the things being integrated must share the meaning of their terminologies.

Finally, it is worthwhile to mention that Pardo and colleagues (2010) conducted a systematic literature review on the same topic, with a similar scope. They have searched for papers on initiatives concerning the harmonization of multiple standards, selecting 32 papers from 1993 to 2009. Our mapping aims at presenting a more recent perspective, considering also publications from the last years (2010 to 2016). Moreover, as discussed, our scope is restricted to those papers using a common model. Pardo's study helped us in some definitions (such as the terms used in the searches and the techniques classification) and showed some possible trends. As a related work, it is further discussed in Section 3.6.

This chapter is organized as follows. Section 3.2 describes the adopted research method, including the research questions and other parameters established for reaching the mapping goal. Section 3.3 presents the mapping results. Section 3.4 discusses the main findings obtained from the results and their implications. Section 3.5 points out the mapping limitations. Section 3.6 discusses related works. Section 3.7 concludes the chapter.

3.2 Research Method

The research method adopted in this mapping study is based on the process defined by Kitchenham and Charters (2007), which has three main phases:

(i) **Planning**: focus on defining a protocol for the study, establishing the research questions, inclusion and exclusion criteria, search strategies, sources of papers, and search string, among others;

(ii) **Conduction**: regards searching and selecting the papers to extract and synthesize information from them;

(iii) **Reporting**: aims at describing the results and distributing them to the interested parties.

This section discusses the main methodological aspects addressed by this mapping, including the research questions, the papers selection, data extraction and synthesis, and classification scheme.

3.2.1 Research Questions

This mapping study addresses the research on harmonization of SE standards. Its purpose is to present a panorama of the current status of the topic. Table 3.1 shows the research questions to be answered and the rationale behind them.

Table 3.1. Research Questions and their Rationales.

No.	Research Question	Rationale
RQ1	When and where have the papers been published?	This RQ focus on revealing when the papers have been published and whether there is any concentration regarding the publication vehicles and venues. This is important to understand the publication trends regarding periods or venues.
RQ2	What are the harmonization perspectives applied?	The perspective applied in the papers refers to what is being harmonized. Some papers focus on the standards' *structure*, usually making correspondences only between structural elements, other focus on the standards' *contents*, dealing with the represented knowledge.
RQ3	What are the harmonization techniques applied?	Along the last years, many different techniques have been proposed to deal with standards harmonization (such as *homogenization*, *comparison*, and *integration*). This RQ aims at identifying the technique(s) applied by each paper.
RQ4	What are the harmonization subject areas?	The scope of this research is the SE area. However, within this area, the standards focus on many different domains, such as *Software Process*, *Information Technology Governance*, *Software Measurement* and so on. This RQ aims at identifying the main areas / domains that are subject of the harmonization efforts.
RQ5	What are the main standards being harmonized?	This RQ aims at identify the main standards used as subject of the harmonization efforts.
RQ6	What are the types of the common conceptual models used in the approaches?	This RQ aims at capturing the type of the common conceptual models adopted by the approaches (such as UML model, metamodel, or ontology).
RQ7	What is the focus of the harmonization initiative?	The standards harmonization efforts have different motivations (such as supporting multiple-model deployment in organizations, supporting standards improvements, or research purposes). This RQ aims at identifying if the focus of each paper is on standards' *developers* or *users*.
RQ8	What types of research have been done?	Scientific papers use to present different types of research (e.g., *solution proposal*, *validation research* and *evaluation research*). This RQ examines the types of research each selected paper presents, helping to understand the maturity of the publications and of the topic.

3.2.2 Paper Selection

As pointed out by Kitchenham and Charters (2007), before selecting the papers, we have to define: (i) search string; (ii) search strategy; (iii) criteria for paper inclusion and exclusion; and (iv) how to store data. These aspects are discussed in the following.

3.2.2.1 Search String

The search string, presented in Table 3.2, is organized in four groups of terms according to our search scope. It was built following two strategies that are pointed by Petersen et al. (2015) as relevant strategies for defining the search string, namely: (i) deriving keywords from known papers, and (ii) iteratively improving the string to find more relevant papers. We have used some already known and relevant publications as control papers to help determining when the string was good enough.

Table 3.2. Search Terms of the Mapping Study.

Search subjects (groups of terms)	Search terms
Subject (Standards)	"standards" OR "reference models"
Approach (Harmonization)	"harmonization" OR "harmonizing" OR "harmonized" OR "integration" OR "integrating" OR "integrated" OR "unification" OR "unifying" OR "unified" OR "combination" OR "combining" OR "combined" OR "mapping" OR "mapped" OR "interoperability" OR "interoperable"
Area (Software Engineering)	"software engineering" OR "software process" OR "process improvement" OR "SPI"
Means (Common Model)	"ontology" OR "ontologies" OR "metamodel" OR "meta-model" OR "common foundation" OR "common model" OR "integration model" OR "reference model"

Search String:
("standards" OR "reference models") AND
("harmonization" OR "harmonizing" OR "harmonized" OR "integration" OR "integrating" OR "integrated" OR "unification" OR "unifying" OR "unified" OR "combination" OR "combining" OR "combined" OR "mapping" OR "mapped" OR "interoperability" OR "interoperable") AND
("software engineering" OR "software process" OR "process improvement" OR "SPI") AND
("ontology" OR "ontologies" OR "metamodel" OR "meta-model" OR "common foundation" OR "common model" OR "integration model" OR "reference model")

The first group of terms establishes the **subject** of the searched papers, in this case, *standards* and *reference models*. The used terms are in plural to return papers working with multiple standards. The second group defines the **approaches** used, and we have a diversity of terms (e.g. *harmonization*, *integration*, *unification*, *combination*, *mapping* and *interoperability*) and their variants as shown in Table 3.2. The third group regards the application **area**, *software engineering*, also involving other terms such as *software process*, *process improvement* and *SPI*. Finally, the fourth group concerns the **means** used to support the harmonization, including terms such as *common model*, *integration model*, *metamodel* and *ontology*.

The search string was applied considering three metadata fields: *title*, *abstract* and *keywords*. Depending on the particularities of each source, we have performed some syntactic adaptations.

3.2.2.2 Search Strategy

The main search strategy we adopted was database search. We selected seven electronic databases, which have shown as the most important from other studies we have previously done, namely:

- ACM Digital Library (http://dl.acm.org);
- Compendex (http://www.engineeringvillage2.org);
- IEEE Xplore (http://ieeexplore.ieee.org);
- Science Direct (http://www.sciencedirect.com);

- Scopus (http://www.scopus.com);
- Springer Link (http://www.springerlink.com);
- Web of Science / ISI of Knowledge (http://www.isiknowledge.com).

To complement the databases searches, we have applied other two search strategies. As suggested by Kitchenham and Charters (2007), we applied backward snowballing to identify additional important publications from the reference lists of the selected papers (JALALI; WOHLIN, 2012). Additionally, manual searches for papers published by relevant researchers and research groups were performed, as suggested by Kitchenham et al. (2011). These researchers / research groups are the authors of the previously selected publications. In this manner, we sought for a broader coverage of the literature, alleviating possible restrictions given by the use of a particular set of electronic databases.

3.2.2.3 Inclusion and Exclusion Criteria

For supporting the selection of the proper papers, we have applied one inclusion criterion (IC) and five exclusion criteria (EC).

The only inclusion criterion is:

IC1 *The paper addresses harmonization of SE standards by means of a common conceptual model.*

The five exclusion criteria are:

EC1 *The paper does not have an abstract;*

EC2 *The paper is just published as an abstract;*

EC3 *The paper is a secondary study, a tertiary study, a summary, a tutorial or an editorial;*

EC4 *The paper is not written in English;* and

EC5 *The paper is an older version of other paper already considered.*

3.2.2.4 Data Storage

Each paper returned during the search phases was properly cataloged and stored. We have used a form in a spreadsheet to organize the relevant information from the identified papers (including: id, source, publication year, title, authors, abstract, keywords and venue) and decisions regarding selecting them or not (criteria applied, comments and analysis). Figure 3.1 illustrates part of this form. We have also detached the search terms (see highlighted terms in Figure 3.1) to help analyzing title and abstract. This form was used for supporting the classification and analysis activities.

ID	Source	Year	Title	Authors	Abstract
#085	Scopus Eng. Village	2008	An **ontology** for quality **standards integration** in software collaborative projects	Ferchichi, Anis(1,2	The objective is to integrate several process using a common **ontology** offering various viewpoints. This methodology is applied to the **integration** of two quality **standards** - ISO 9001:2000 and CMMI - in order to generate a multivues quality **ontology** allowing a double certification relative to these two **standards**. This work is carried out within a **software engineering** company (Sylis). Human and cultural aspects of the company are
#133	Scopus Eng. Village Springer	2012	**Standards harmonization**: Theory and practice	Henderson-Sellers	As **software engineering** (and other) **standards** are developed over a period of years or decades, the suite of **standards** thus developed often begins to lose any cohesion that it originally possessed. This has led to discussions in the **standards** communities of possible collaborative development, **interoperability** and **harmonization** of their existing **standards**. Here, I assess how such **harmonization** efforts may be aided by recent research
#142	Scopus Eng. Village WoScience Sc. Direct ACM	2012	An **ontology** for the **harmonization** of multiple **standards** and models	Pardo, César(1,2);	**Harmonization** plays an important role in organizations that are seeking to resolve manifold needs at their different hierarchical levels through multiple models such as CMMI, ISO 90003, ITIL, SWEBOK, COBIT, amongst others. A great diversity of models involves a wide heterogeneity not only about structure of their process entities and quality systems, but also with regards to terminology. This article presents an **ontology**

Figure 3.1. Data Storage Form (fragment).

3.2.2.5 Assessment

The mapping protocol was tested to verify its viability and adequacy. A set of preselected papers considered germane to our mapping was used as control group. For establishing the search string, the set of search terms was refined iteratively, starting from a primary string and improving it in iterations until all papers in the control group return. The selection activities were performed by the doctorate candidate and reviewed by the advisors. In the first selection stage (selection by *title*, *abstract* and *keywords*), from the returned papers, they have reviewed all the selected ones and a sample of 30% of the discarded ones. From the second stage on, all the decisions were reviewed. To reduce bias, the divergent classifications were discussed to achieve a consensus and to improve classification precision.

3.2.3 Data Extraction and Synthesis

The search process comprises five stages, being the first three related to searches in the databases, as follows: (1) Duplication removal, (2) Selection by *title*, *abstract* and *keywords*, (3) Selection by the *full text*, (4) Snowballing and (5) Search by Researchers / Research Groups.

Searches in the databases were conducted in two rounds. The first one, in September 2015, searched in the aforementioned seven bases, finding 877 publications of which 545 where distinct. By analyzing the *title*, *abstract* and *keywords*, 43 papers were selected. Finally, by analyzing the *full text*, we have selected 19 papers. In January 2017, we updated the searches, considering four bases: *Scopus*, *Compendex*, *Web of Science* and *Science Direct*. The searches in the other three bases were not updated due to the fact that, in the first round, none of those three bases has exclusively selected papers in the final results. Thus, considering the cost-benefit relation, we considered that it was not worth updating the search in them.

Considering both search rounds, we reached a total of **1050 publications**, being 491 from *Scopus*, 255 from *Compendex*, 60 from *Web of Science*, 36 from *Science Direct* (the four sources with updated searches), and 129 from *IEEE Xplore*, 43 from *ACM Digital Library*, and 36 from *Springer Link*. Figure 3.2 shows the final numbers of publications retrieved and the selection stages.

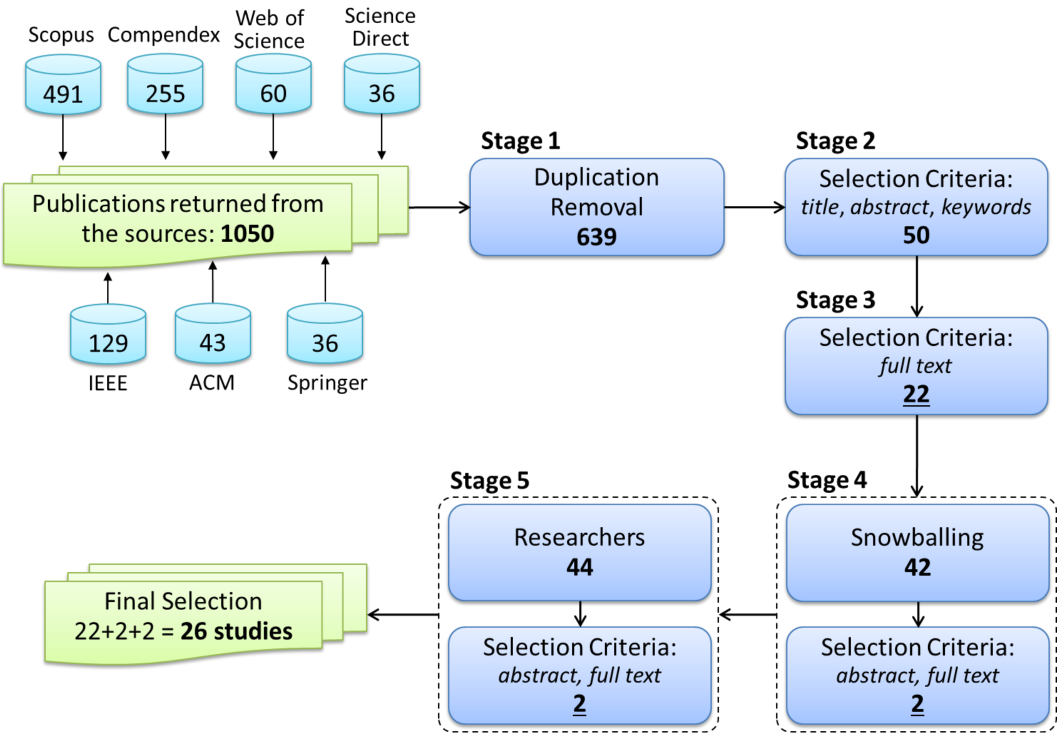


Figure 3.2. Search and Selection Process.

The **1st stage** removed duplications (publications appearing in multiple sources), achieving **639 publications** (39% of reduction).

The **2nd stage** applied the selection criteria (for inclusion and exclusion) over *title*, *abstract* and *keywords*, resulting in **50 papers** (reduction of 92%): 2 papers were removed by EC1 (*The paper does not have an abstract*), 40 by EC3 (*The paper is a secondary study, a tertiary study, a summary, a tutorial or an editorial*), and 547 for not satisfying IC1 (*The paper addresses harmonization of SE standards by means of a common model*). At this stage, as suggested by Kitchenham and Charters (2007), we only eliminated publications clearly unrelated to the subject. In case of doubt, the paper was taken to the next stage.

The **3rd stage** applied the selection criteria considering the *full text*, reaching a set of **22 papers** (reduction of 56%): 3 papers were eliminated by EC4 (*The paper is not written in English*), 8 by EC5 (*The paper is an older version of another paper already considered*), and 17 papers were excluded for not satisfying IC1 (*The paper addresses harmonization of SE standards by means of a common model*).

In the **4th stage**, we performed snowballing from the 22 selected publications. We have searched in the papers' texts for references to other possibly relevant publications (backward snowballing). We have found 58 distinct related publications, being 19 already analyzed in the previous stages. Firstly, the selection criteria were applied over *title*, *abstract* and *keywords*, remaining 11 papers (72% of reduction over the 39 papers analyzed). For these 11 papers, the selection criteria were applied over the *full text*, and only **2 papers** remained (reduction of 82%). We have recursively applied snowballing for these two papers, finding 3 new publications, but none was selected. From the 42 papers analyzed in this stage, 30 were eliminated for not satisfying IC1 (*The paper addresses harmonization of SE standards by means of a common model*), 8 by EC3 (*The paper is a secondary study, a tertiary study, a summary, a tutorial or an editorial*), and 2 by EC5 (*The paper is an older version of another paper already considered*).

Finally, in the **5th stage**, we manually searched for papers from the researchers and research groups involved in the already selected publications. To conduct that, we searched in *Scopus* (the digital library that returned more papers, including 21 out of the 22 selected in the 3rd stage), in the *DBLP Computer Science Bibliography* and in the author profile in *Google Scholar* (for diversifying the searches). Still adopting the Kitchenham and Charters (2007) recommendation, we selected all papers seeming related to the mapping scope. From all the authors searched, 95 distinct papers were identified by the *title*, being 51 already analyzed (42 in the first stages and 9 in the snowballing). After applying the selection criteria over *abstract* and *keywords*, 15 papers remained (reduction of 66% over the 44 papers analyzed). For these 15 papers, the selection criteria were applied considering the *full text*, and only **2 papers** remained (reduction of 87%). The recursive search on the publications of the authors of these two papers has not returned relevant new results. From the 44 papers analyzed in this stage, 34 were eliminated for not satisfying IC1 (*The paper addresses harmonization of SE standards by means of a common model*), 4 by EC3 (*The paper is a secondary study, a tertiary study, a summary, a tutorial or an editorial*), 2 by EC4 (*The paper is not written in English*), and 2 by EC5 (*The paper is an older version of another paper already considered*).

At the end of the five stages, we have selected **26 papers** to be analyzed (22 from the sources, 2 from snowballing, and 2 from manual search to researchers / research groups). Table 3.3 presents the stages and their main results in numbers. As we can see, the selection process caused a progressive reduction of the considered publications through the selection process stages.

Table 3.3. Selection Stages Results.

Stage	Applied criteria	Analyzed content	Initial number of papers	Final number of papers	Reduction (%)
1st (Sources)	Duplicate Removal	Title, abstract	1050	639	39%
2nd (Sources)	IC1, EC1, EC2, EC3	Title, abstract, keywords	639	50	92%
3rd (Sources)	IC1, EC4, EC5	Full Text	50	**22**	56%
4th (Snowballing)	IC1, EC3	Title, abstract, keywords	42	11	74%
	IC1, EC5	Full Text	11	**2**	82%
5th (Researchers)	IC1, EC3, EC4	Title, abstract, keywords	44	15	66%
	IC1, EC4, EC5	Full Text	15	**2**	87%
Final Result	-	-	639 (sources) + 42 (snowballing) + 44 (authors) = **725**	22 (sources) + 2 (snowballing) + 2 (authors) = **26**	96,4%

Table 3.4 presents the bibliographic reference of the 26 selected papers with an identifier [#00] for each paper, ordered by publication year.

Table 3.4. Selected Papers.

ID	Bibliographic Reference
#01	Lepasaar, M., and Mäkinen, T. (2002). Integrating software process assessment models using a process meta model. In *Engineering Management Conference*. IEMC'02. IEEE International, vol. 1, 224-229.
#02	Liao, L., Qu, Y. and Leung, H. (2005). A software process ontology and its application. In *Proceedings of the First International Workshop on Semantic Web Enabled Software Engineering*.
#03	García, F., Bertoa, M., Calero C., Vallecillo, A., Ruíz, F., Piattini, M., and Genero, M. (2006). Towards a consistent terminology for software measurement. *Information and Software Technology* 48(8), 631-644.
#04	Mäkinen, T. and Varkoi, T. (2007). A harmonized design for process assessment indicators. *Software Process: Improvement and Practice* 12(4), 331-338.
#05	Ferchichi, A., Bigand, M., and Lefebvre, H. (2008). An ontology for quality standards integration in software collaborative projects. In *Proceedings of the First International Workshop on Model Driven Interoperability for Sustainable Information Systems (MDISIS'08)*, 17–30, Montpellier, France.
#06	Salviano, C.F. and Figueiredo, A.M.C. (2008). Unified basic concepts for process capability models. In *International Conference on Software Engineering and Knowledge Engineering (SEKE'2008)*, 173-178.
#07	Falbo, R.A. and Bertollo, G. (2009). A software process ontology as a common vocabulary about software processes. *Int. Journal of Business Process Integration and Management* 4(4), 239-250.
#08	Lochmann, K. and Goeb, A. (2011). A unifying model for software quality. In *Proceedings of the 8th International Workshop on Software Quality*, 3-10. ACM.
#09	Pardo, C., Pino, F.J., García, F., Piattini, M. and Baldassarre, M.T. (2012). An ontology for the harmonization of multiple standards and models. *Computer Standards & Interfaces*, 34(1), 48-59.
#10	Baldassarre, M.T., Caivano, D., Pino, F.J., Piattini, M. and Visaggio, G. (2012). Harmonization of ISO/IEC 9001:2000 and CMMI-DEV: from a theoretical comparison to a real case application. *Software Quality Journal*, 20(2), 309-335.

ID	Bibliographic Reference
#11	Banhesse, E.L., Salviano, C.F. and Jino, M. (2012). Towards a metamodel for integrating multiple models for process improvement. In *38th Euromicro Conference on Software Engineering and Advanced Applications*. 315-318. IEEE.
#12	Rahman, A.A., Sahibuddin, S. and Ibrahim, S. (2012). Using taxonomy comparative analysis for the unification of process improvement frameworks. *International Journal of Digital Content Technology and its Applications*, 6(21), 34.
#13	Henderson-Sellers, B. (2012). Standards harmonization: theory and practice. *Software & Systems Modeling*, 11(2), 153-161.
#14	Jeners, S., Lichter, H. and Dragomir, A. (2012). Towards an integration of multiple process improvement reference models based on automated concept extraction. In *European Conference on Software Process Improvement*. Springer Berlin Heidelberg, 205-216.
#15	Jeners, S., Lichter, H. and Pyatkova, E. (2012). Metric based Comparison of Reference Models based on Similarity. *International Journal of Digital Content Technology and its Applications*, 6(21), 50.
#16	Jeners, S. and Lichter, H. (2013). Smart integration of process improvement reference models based on an automated comparison approach. In *European Conference on Software Process Improvement*. 143-154. Springer Berlin Heidelberg.
#17	Jeners, S., Lichter, H. and Rosenkranz, C.G. (2013). Efficient Adoption and Assessment of Multiple Process Improvement Reference Models. *e-Informatica Software Engineering Journal*, 7(1).
#18	Pardo, C., Pino, F.J., García, F., Baldassarre, M.T. and Piattini, M. (2013). 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), 125-143.
#19	Barcellos, M.P. and Falbo, R.A. (2013) A software measurement task ontology. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. 311-318. ACM.
#20	Pardo, C., García, F., Piattini, M., Pino, F.J. and Baldassarre, M.T. (2014). A reference ontology for harmonizing process-reference models. *Revista Facultad Ingeniería Universidad Antioquia*, (73), 29-42.
#21	Henderson-Sellers, B., Gonzalez-Perez, C., McBride, T., Low, G. (2014). An ontology for ISO software engineering standards: 1) Creating the infrastructure. *Computer Standards & Interfaces*, 36(3), 563-576.
#22	Ruy, F.B., Falbo, R.A., Barcellos, M.P., Guizzardi, G. and Quirino, G.K. (2015). An ISO-based software process ontology pattern language and its application for harmonizing standards. *ACM SIGAPP Applied Computing Review*, 15(2), 27-40.
#23	Pardo. C., García, F., Piattini, M., Pino, F.J. and Baldassarre, M.T. (2015). A 360-degree process improvement approach based on multiple models. *Revista Facultad de Ingeniería Universidad de Antioquia*, (77), 95-104.
#24	Pardo, C., García, F., Piattini, M., Pino, F.J., Lemus S. and Baldassarre, M.T. (2016). Integrating Multiple Models for Definition of IT Governance Model for Banking ITGSM. *International Business Management*, 10, 4644-4652.
#25	Gonzalez-Perez, C., Henderson-Sellers, B., McBride, T., Low, G.C. and Larrucea, X. (2016). An ontology for ISO software engineering standards: 2) Proof of concept and application. *Computer Standards & Interfaces*, 48, 112-123.
#26	Mejia, J., Muñoz, E. and Muñoz, M. (2016). Reinforcing the applicability of multi-model environments for software process improvement using knowledge management. *Science of Computer Programming*, 121, 3-15.

Table 3.5 groups these papers by the main research group conducting the harmonization efforts [G00]. Henceforth these identifiers are used to refer to the corresponding paper [#00] or to the groups' initiative [G00].

Table 3.5. Papers Grouped by Research Group.

ID	Group Name	Paper ID	Year
G01	Tampere University of Technology, Finland	#01	2002
		#04	2007
G02	Southeast University, Nanjing, China	#02	2005
G03	Alarcos, Universidad Castilla-La Mancha - UCLM, Spain	#03	2006
		#09	2012
		#10	2012
		#18	2013
		#20	2014
		#23	2015
		#24	2016
G04	LGIL, Ecole Centrale de Lille, France	#05	2008
G05	CTI Renato Archer / Unicamp, Brazil	#06	2008
		#11	2012
G06	NEMO, Federal University of Espírito Santo - UFES, Brazil	#07	2009
		#19	2013
		#22	2015
G07	University of Technology Munich, Germany	#08	2011
G08	UNIKL / Universiti Teknologi Malaysia - UTM, Malaysia	#12	2012
G09	University of Technology Sydney - UTS, Australia	#13	2012
		#21	2014
		#25	2016
G10	RWTH Aachen University, Germany	#14	2012
		#15	2012
		#16	2013
		#17	2013
G11	CIMAT, National Council of Science and Technology, Mexico	#26	2016

As we can observe, the 26 selected papers were produced by 11 research groups from universities and research centers in diverse countries. Although some groups have a more expressive participation, the topic is not concentrated or dominated by any specific group, but subject of research by several distinct efforts.

3.2.4 Classification Scheme

The classification scheme should consider different facets, one for each research question (PETERSEN et al., 2008). For deriving the classification scheme, we followed two different approaches: (i) using categories already considered in the literature (the case of RQ1 and RQ8); and (ii) considering categories emerging from the selected papers (for RQ2 to RQ7). Following, the categories considered for each facet are discussed.

3.2.4.1 Harmonization Perspectives (RQ2)

A relevant information to be collected from the papers refers to on what perspective(s) the standards are being harmonized. Some publications focus on the standards' structure, making correspondences only between structural elements, those used to organize / categorize the standards' contents (such as *process area*, *goal* and *practice* in CMMI-DEV; or *process*, *activity* and *task* in ISO/IEC 12207); others focus on the standards' contents, dealing with the represented knowledge (such as *Requirements Engineering* (a process area), *Manage Requirements* (a goal) and *Obtain an Understand of Requirements* (a practice) in CMMI-DEV). In this facet, we want to capture if the harmonization perspective focuses on the **Standards' Structure**, on the **Standards' Content**, or on both.

3.2.4.2 Harmonization Techniques (RQ3)

Along the years, some techniques have been proposed and improved to deal with standards harmonization. This facet aims at identifying the technique(s) applied by each paper. Based on the classification provided by Pardo and colleagues (2012), we consider the following categories:

- **Homogenization**: refers to the adaptation of each standard to a predefined structure to put different standards in a homogeneous form (same structure). This technique is related to the *structural* perspective, and many times is used to support other techniques. By homogenization, the elements of each standard are classified into a common referential structure being more easily handled.
- **Comparison / Mapping**: Comparison analyzes standards or their processes for similarities and differences. Mapping focuses on comparing lower-level standards' elements (such as activities, work products and roles) establishing a link between them. Concerning the perspectives, the structural comparison / mapping of multiple standards makes the correspondences between only structural elements (e.g., *process* x *process area*, *artifact* x *document*) while the content comparison / mapping deals only with their content elements (e.g., *Project Management* x *Project Planning*, *Product Assurance* x *Product Objective Evaluation*, *Project Team* x *Project Staff*).
- **Integration / Unification**: relates to the definition of a new (integrated) model unifying selected practices / definitions of different standards. The structural integration regards the production of a (structural) model representing a unified view of the structures of multiple standards, whilst content integration results in a (content) model unifying the standards' contents.

3.2.4.3 Harmonization Subject Areas (RQ4)

The scope of this research is the SE area. However, within this area, the standards may focus on many different domains, such as *Software Process*, *Information Technology Governance*, *Software*

Measurement, *Requirements* and so on. In this facet, we identify the domains used as subject of the harmonization efforts.

3.2.4.4 Standards being Harmonized (RQ5)

The harmonization efforts have been applied on diverse standards. In this facet, we want to capture the standards (such as *CMMI-DEV*, *ISO/IEC 12207*, *ISO/IEC 15504*, etc.) target of the harmonization efforts.

3.2.4.5 Types of Common Model Used (RQ6)

Since the mapping scope requires the use of a common conceptual model for harmonization, in this facet, we want to capture the type of the common conceptual models adopted by the approaches. We have identified three main categories of such types:

- **Metamodel**: represents a common structure of each or a set of standards, used for instantiating specific domain models.
- **Ontology**: represents the knowledge of the subject domain or area, not necessarily committed to any specific standard.
- **Classification Model**: represents a set of terms, typically organized in a hierarchy, applied to classify / categorize the standards' elements (such as a Taxonomy or Folksonomy).

It is important to say that this is not a trivial classification. By analyzing the papers, it is possible to observe a lack of rigor when the authors classify the common models. For example, the same model is called *metamodel* and *ontology* in different or even in the same paper. Indeed, considering the information available in some papers, it could be hard to precisely define the right type of model used. Thus, to preserve the study repeatability, we have used the terms adopted by the papers' authors, in preference to a further discussion on how each common model could be categorized.

3.2.4.6 Harmonization Results Focus (RQ7)

The standards harmonization efforts have different motivations (such as supporting multiple model deployment in organizations, supporting standards improvements, or providing a common vocabulary / conceptualization). These motivations conduct to results usually focused on **Standards' Users** (such as software organizations) or on **Standards' Developers** (standardization organizations). In this facet, we want to capture the papers declared focus.

3.2.4.7 Research Types (RQ8)

This research question is common to several mapping studies, as pointed out by Petersen et al. (2015). Thus, we have adopted the classification proposed by Wieringa et al. (2006), and revisited by Petersen et al. (2015) to become more general. This research question helps to understand the maturity stage and weight of the publications and of the topic as a whole. We have considered only three categories, those applicable for at least one of the selected papers, namely:

- **Solution proposal**: research approach in which the publication proposes a solution for a problem and argues for its relevance, independently of a complete validation. The solution can be either a novel or a significant extension of an existing one. The potential benefits and the applicability of the solution is shown by a small example or a good line of argumentation.
- **Validation research**: the publication investigates the characteristics of a proposed solution not yet implemented in practice. It may have been previously proposed elsewhere. The investigation is carried out by a research method, which can include prototyping, simulation, academic case studies and experiments.
- **Evaluation research**: the publication discusses the implementation and evaluation of a proposed solution in an industrial setting. It shows how the techniques are implemented in practice (solution implementation) and what are the consequences of the implementation in terms of benefits and drawbacks (implementation evaluation). The techniques' novelty is not necessarily a contribution of the study. However, some kind of industry cooperation or real world project is needed (PETERSEN et al., 2008).

It is worth saying that although papers categorized only as Solution Proposal provide no empirical evidence (WIERINGA et al., 2006) and, thus, should not be considered in systematic reviews, according to Petersen et al. (2015), they are important in a mapping study to spot trends of topics being worked on. Thus, in our study we have also considered papers that only present solution proposals.

3.3 Results

The mapping study was performed following the steps described in Section 3.2. This section presents the results for the research questions defined in Section 3.2.1 (RQ1 to RQ8).

3.3.1 Classification by Publication Year and Source (RQ1)

A general view of the publications on SE Standards Harmonization is shown in Figure 3.3. It presents the 26 selected papers distributed over the publication vehicles and years. As this figure shows, the harmonization initiatives applying a common conceptual model are relatively recent, starting essentially in 2002, and with few occurrences during the 2000's. In the beginning of the 2010's, the number of publications increased, with the main identified research groups involved and producing more mature results, as discussed in Section 3.4. Actually, this interest growth seems to be started in 2011, since some borderline papers excluded by EC5 (*The paper is an older version of another paper* *already considered*) are from that year. In the selection process, we have discarded 12 publications by EC5: 5 spread from 2006 to 2010, 4 from 2011, and 3 from 2012 to 2015.

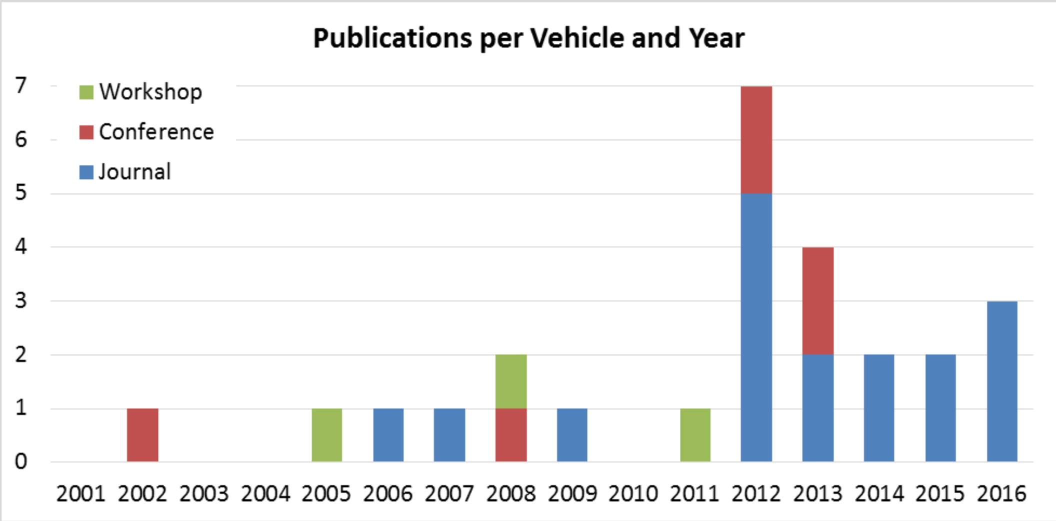


Figure 3.3. Distribution of the Selected Papers over the Years and Vehicles.

Three main vehicles have published the selected papers: Journals, Conferences and Workshops. Journals have been the main forum for publishing the harmonization initiatives, covering 65% (17 out of 26 papers), and all papers in the last three years. Conferences are the publication forum of 23% (6 out of 26). Finally, three papers were presented in Workshops (12%). The 26 papers were published in 21 different venues (being 1 venue with 3 papers, and 3 venues with 2 papers each), meaning that this topic has not a preferred venue yet, although the *Computer Standards & Interfaces Journal* stands out. Table 3.6 presents the publication sources of the selected papers, their types and papers' IDs. Publications vehicles in areas such as Software Engineering, Software Process, and Standards seem to be more receptive for the studied papers.

Table 3.6. Publication Sources.

Publication source	Paper ID
Journals	
Computer Standards & Interfaces	#09, #21, #25
International Journal of Digital Content Technology and its Applications	#12, #15
Revista Facultad de Ingeniería Universidad de Antioquia	#20, #23
ACM SIGAPP Applied Computing Review	#22
e-Informatica Software Engineering Journal	#17
Information and Software Technology	#03
International Business Management	#24
International Journal of Business Process Integration and Management	#07
Journal of Systems and Software	#18
Science of Computer Programming	#26
Software & Systems Modeling	#13
Software Process: Improvement and Practice	#04
Software Quality Journal	#10
Conferences	
European Conference on Software Process Improvement	#14, #16
Annual ACM Symposium on Applied Computing	#19
Engineering Management Conference	#01
International Conference on Software Engineering and Knowledge Engineering	#06
Software Engineering and Advanced Applications	#11
Workshops	
International Workshop on Model Driven Interoperability for Sustainable Information Systems	#05
International Workshop on Semantic Web Enabled Software Engineering	#02
International Workshop on Software Quality	#08

3.3.2 Harmonization Perspectives (RQ2)

Table 3.7 shows the papers adopting: (i) a structural perspective, focusing on the structural aspects of the standards, (ii) a content perspective, focusing on the knowledge contents provided by the standards, or (iii) both. For example, Lepasaar and Mäkinen (2002) [#01] propose a metamodel with general concepts such as *activity*, *resource* and *artifact*, and maps it to the structural elements of CMMI and ISO/IEC 15504 (such as *process area*, *practice* and *work product*). In turn, Li et al. (2005) [#02] propose an ontology that, besides the structural elements, also compares content elements (such as *Project Planning*, *Requirements Development* and *Configuration Management*, all instances of the CMMI's *process area* structural element).

As Figure 3.4 shows, from the 26 papers, 8 (31%) adopted a pure *Structural* perspective, 2 (8%) adopted a pure *Content* perspective, and 16 (61%) focused on both perspectives. The most relevant trend perceived in this facet is that publications before 2012 have often focused on a single perspective (63%), whilst the papers from 2012 onwards mostly focused in both perspectives (72%).

Table 3.7. Harmonization Perspectives.

Perspective	Paper ID
Structure (only)	#01, #04, #07, #08, #11, #13, #20, #21
Contents (only)	#03, #19
Structure + Contents	#02, #05, #06, #09, #10, #12, #14, #15, #16, #17, #18, #22, #23, #24, #25, #26

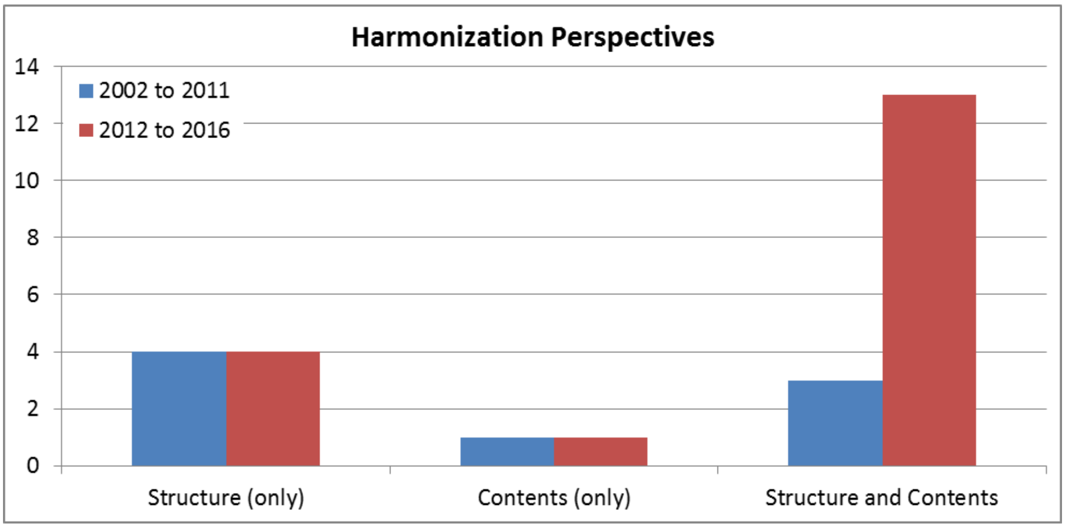


Figure 3.4. Harmonization Perspectives by Period.

3.3.3 Harmonization Techniques (RQ3)

Three categories of techniques were identified in the papers: Homogenization, for adapting each standard to a predefined structure; Comparison / Mapping, for making the correspondences between structural or content elements (or both); and Integration / Unification, for producing a unified view of the involved standards considering their structure or content (or both). Table 3.8 presents the techniques adopted by each paper; Figure 3.5 shows in which papers they are combined; and Figure 3.6 presents it in a time perspective (with the number of publications inside the circles). Publications applying *Homogenization* (81%) usually do that for supporting another technique. *Integration* techniques (present in 38% of the papers) show a subtle growth in their adoption in the last years, when compared to the *Mapping* ones (present in 73%). Moreover, only 23% (6 papers: #05, #09, #17, #18, #23, #24) used the three techniques together, combining them in a more elaborated harmonization approach. However, despite of the presented data, the papers have shown an evolution of the techniques along the years. New resources, such as similarity metrics, concepts extraction, ontologies, patterns and model derivation, have been used for improving the techniques, reaching more advanced results in standards harmonization.

Table 3.8. Harmonization Techniques.

Harmonization Technique		Paper ID
Homogenization	Structural	#02, #05, #06, #08, #09, #10, #11, #12, #13, #14, #15, #16, #17, #18, #20, #21, #22, #23, #24, #25, #26
Comparison / Mapping	Structural	#01, #07, #11, #12, #14, #16, #17
	Contents	02, #03, #05, #06, #09, #10, #12, #15, #16, #17, #18, #19, #23, #24, #26
Integration / Unification	Structural	#04, #09, #18, #21, #22, #25
	Contents	#05, #09, #17, #18, #22, #23, #24, #25

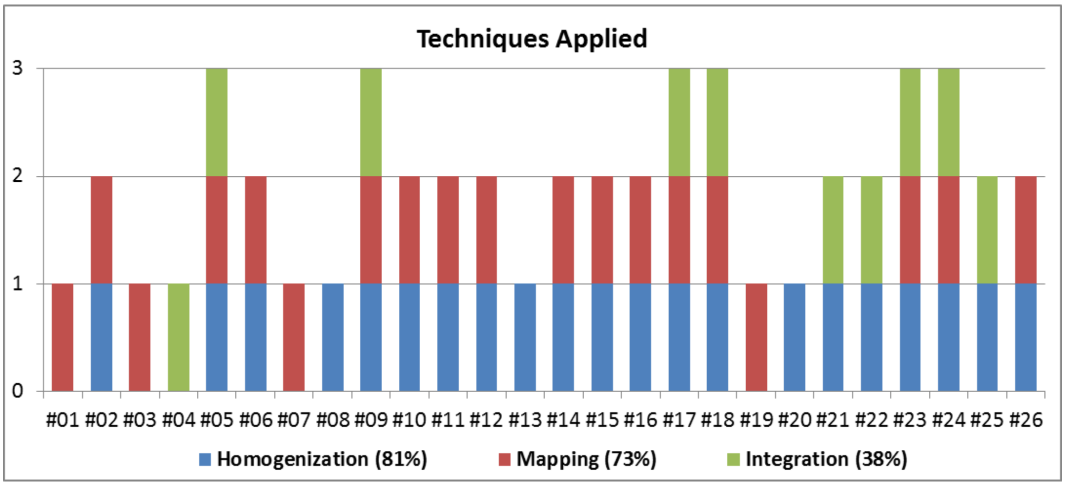


Figure 3.5. Techniques applied by Paper.

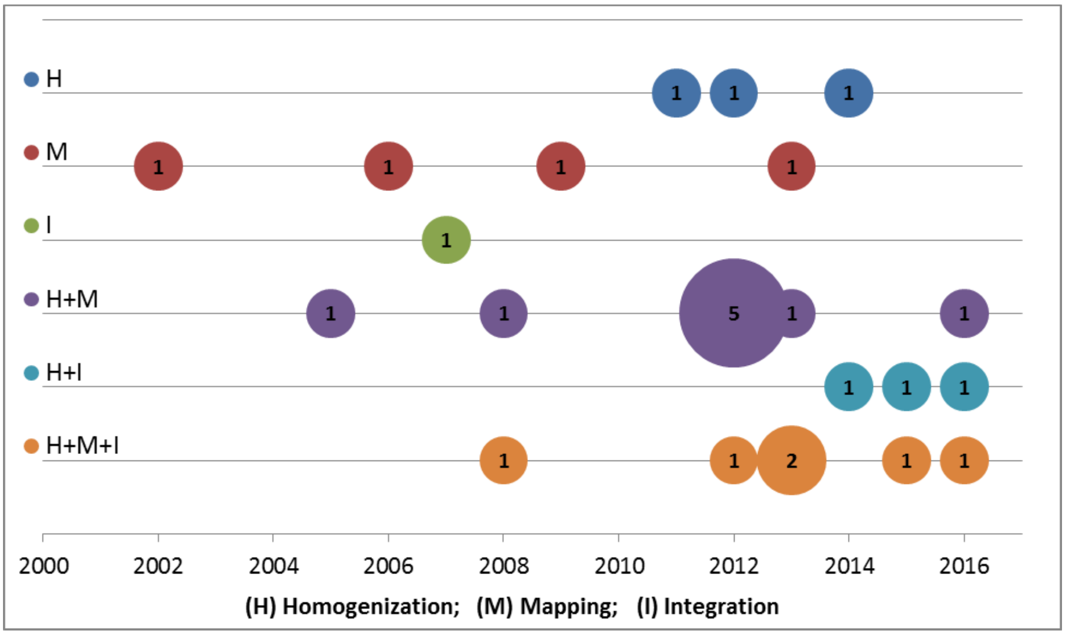


Figure 3.6. Techniques applied over the Years.

3.3.4 Harmonization Subject Areas (RQ4)

Considering the subject areas for harmonization, as Table 3.9 shows, there is a clear focus on harmonizing *Software Process* standards, addressed by 81% of the papers, mainly from 2012 on. It is the most general area, with more standards and adoption, thus it is natural to see more publications on that. *IT Governance* has some focus, being subject of 38% of the papers (a total of ten papers, nine of them focusing also on *Software Processes*). Finally, more specific subjects such as *Software Measurement*, *Process Assessment Indicators* and *Software Product* were applied, exclusively, by four papers (15%).

Table 3.9. Harmonization Subject Areas along the Years.

Subject Area	**Year (2002-2016)**															**Total**
	02	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	
Software Process	#01			#02			#05, #06	#07			#09, #10, #11, #12, #13, #14, #15	#16, #17, #18	#20, #21	#22, #23	#25, #26	21 (81%)
IT Governance											#09, #12, #14, #15	#16, #17, #18	#20	#23	#24	10 (38%)
Software Measurement					#03							#19				2 (8%)
Process Assessment Indicators						#04										1 (4%)
Software Product										#08						1 (4%)

3.3.5 Standards being Harmonized (RQ5)

We have identified a total of 34 distinct standards being subject of harmonization in the selected papers. Since the main subject area is Software Process, its main standards get the focus, as Table 3.10 shows. *CMMI* (present in 17 papers, 65%), *ISO/IEC 15504* (11 papers, 42%) and *ISO/IEC 12207* (6 papers, 23%) are among the most focused standards. The other one is an IT Governance standard, *COBIT* (present in 9 papers, 35%). Several other standards were also subject of five or less papers, as the table shows. Considering the number of harmonized standards, the publications presented initiatives typically using two or three standards, but it reached a maximum of six [#18].

Table 3.10. Main Standards being Harmonized.

Subject Standard	Year (2002-2016)															Total
	02	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	
CMMI	#01			#02			#05, #06	#07	#10		#11, #12, #14, #15	#16, #17, #19	#20	#22, #23	#26	17 (65%)
ISO 15504	#01			#02		#04	#06	#07			#11, #12, #13, #15	#16	#21			11 (42%)
COBIT											#09, #14, #15	#16, #17, #18	#20	#23	#24	9 (35%)
ISO 12207								#07			#13	#19	#21	#22	#25	6 (23%)
ISO 9001							#05	#07	#10				#20	#23		5 (19%)
ISO 20000											#09, #12,	#18	#20	#23		5 (19%)
ISO 27001/27002											#09	#18	#20	#23	#24	5 (19%)
ITIL / BASEL / VAL IT / RISK IT											#09	#18	#20	#23	#24	5 (19%)
ISO 24744											#13		#21		#25	3 (12%)
Other*					#03			#07	#10	#08	#13, #14	#17, #19	#21	#22	#25, #26	-

* Other referenced standards have appeared twice (ISO 24765, ISO 15939, ISO 15288, SWEBOK, Functional Safety, SPEM, PSM) or once (MR-MPS-SW, ISO 29110, ISO 14598, ISO VIM, IEEE 610.12, IEEE 1061, RUP, ISO 25010, ABQM, DQM-SOA, OOSPICE, SixSigma, SME Process, GQM).

3.3.6 Types of Common Models Used (RQ6)

Two main types of common models are used by the selected papers, as Table 3.11 and Figure 3.7 show: *Ontologies*, adopted by 16 papers (62%), and *Metamodels*, present in 10 papers (38%). Other types of models, focused only on classification, are used in only one paper each (4%), namely: Taxonomy and Folksonomy. Two papers used more than one type of common model. Observing the period until 2012 (included), from the 15 publications, 8 used metamodels and 6 ontologies (and 1 a taxonomy). After that, ontologies started to play a more significant role, being in 10 out of the 11 papers (91%), while metamodels are used in only 2 (18%), both in 2013. This can be explained by the gain of popularity of ontologies, but also by related resources (such as semantic representation, inferences and ontology patterns) potentially useful for harmonization purposes (PARDO et al., 2012) (HENDERSON-SELLERS et al., 2014) (RUY et al., 2015).

Table 3.11. Types of Common Models Adopted.

Type of Common Model	Paper ID
Ontology	#02, #03, #05, #07, #09, #10, #16, #18, #19, #20, #21, #22, #23, #24, #25, #26
Metamodel	#01, #04, #06, #08, #11, #13, #14, #15, #16, #17
Other	#12, #26

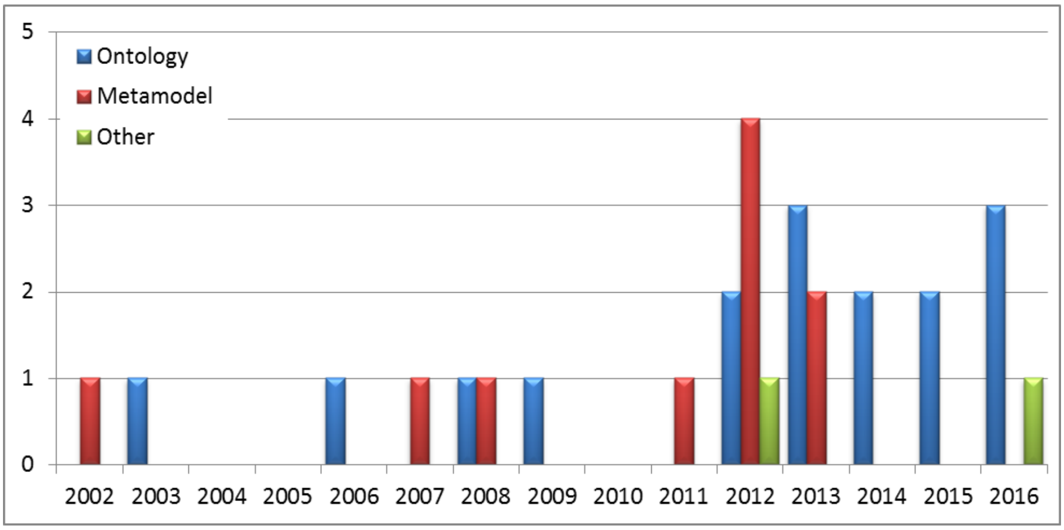


Figure 3.7. Distribution of the Common Model Type adopted over the Years.

3.3.7 Harmonization Results Focus (RQ7)

Considering the focus declared by each publication, as Table 3.12 shows, 22 of them (85%) focus the results on *Standards' Users*, mainly on software organizations, while 6 (23%) focus on *Standards' Developers*, the standardization organizations (such as ISO). The papers focusing on the *Standards' Users* typically concern about the adoption of multiple standards by software organizations. The four papers focusing exclusively on *Standards' Developers* (#13, #21, #22 and #25) have proposed solutions related to the ISO Harmonization Initiative (HENDERSON-SELLERS et al., 2014).

Table 3.12. Harmonization Results Focus.

Harmonization Focus	Paper ID
Standards' Users	#01, #02, #03, #04, #05, #06, #07, #08, #09, #10, #11, #12, #14, #15, #16, #17, #18, #19, #20, #23, #24, #26
Standards' Developers	#03, #04, #13, #21, #22, #25

3.3.8 Research Types (RQ8)

Regarding research types, all 26 papers present a *Solution Proposal*, or part of it complementing another publication (in the case of the groups). As Table 3.13 shows, almost three-quarters of the papers presented some kind of assessment, being 13 (50%) with *Validation Research* and 6 (23%)

with *Evaluation Research*. It shows that most of the efforts discussed in the papers have a concern about assessing the work, but just in few cases it has reached industry or a real world project (5 out of the 6 studies with evaluation research are from the same research group (G03), with the same industry partners).

Table 3.13. Assessment Types.

Assessment Types	Paper ID
None (only Solution Proposal)	#01, #04, #08, #11, #12, #13, #21
Validation Research	#02, #03, #06, #07, #14, #15, #16, #17, #19, #22, #24, #25, #26
Evaluation Research	#05, #09, #10, #18, #20, #23

3.4 Discussion

Software organizations have been adopting quality standards for diverse reasons: for improving their software processes (DAVENPORT, 2005), for interoperating with their clients and partners (GARCÍA et al., 2006), as it is mandatory for certain market niches, and for taking legal benefits from the government (PARDO et al., 2015), among others. These standards are frequently used in combination, in the so called Multi-model Environments (PARDO et al., 2013) (MEJIA et al., 2016). Beyond the synergic advantages expected by the use of multiple standards (FERCHICHI et al., 2008), these organizations can experience also some divergences and incompatibilities related to standards' presentation, vocabulary and meaning.

This scenario has gained the attention of researchers mainly in the last years. From the 26 papers we analyzed, 19 (73%) were published in the current decade (2010's). Another evidence of the recent further exploration of this scenario is given by the concentration of papers from some research groups. We can observe in the groups with larger number of publications (specifically G03, G06 and G09) preliminary efforts on building conceptual models from the standards (for providing a common vocabulary or a shared conceptualization in certain areas) and, then, focusing on techniques and approaches for guiding harmonization initiatives. It points out to a maturation of the harmonization efforts in these groups.

Considering the focus given to the papers' results, most of them (85%) are related to standards' users, such as software organizations, in general proposing solutions for multi-model environments. However, 23% of the selected papers focus on the standards' developers, attempting to solve or mitigate the problem from its source. Some of these works are associated to standardization organizations, such as ISO, revealing joint efforts with academia.

The main area focused is, as expected, Software Processes. It is the subject area for 81% of the papers, working with standards such as CMMI (65%), ISO/IEC 15504 (42%) and ISO/IEC 12207 (23%).

CMMI is the standard present in most publications and also in most research groups (9 out of 11). IT Governance is also representative, being subject of 38% of the papers, concentrated in 3 research groups. The main IT Governance standards focused are COBIT (35%) and ISO/IEC 20000, ITIL, BASEL II, VAL IT and Risk IT (19% each).

Regarding the techniques applied, the majority of the papers propose some kind of Homogenization (81%). Since this technique is used to put the standards' knowledge in a pre-defined structure / format easing comparisons, it is often applied as a previous step for the other techniques. Mapping techniques are applied in 73% of the papers, sometimes for providing the final results of the publication, sometimes as an intermediary step before Integration, which is applied in 38% of the papers. As the research in this topic evolves, some resources / technologies have been used together with these techniques seeking for better results, as it is the case of similarity metrics [#15], (semi-)automatic concepts extraction [#14], modeling patterns [#22], model derivation [#21], knowledge management [#26], and management tools [#18, #26], among others.

Still focusing on the methodological aspects, some research groups have explored a better combination of techniques, frequently resulting on the proposition of a harmonization approach. We have selected for further analysis initiatives from five of these groups with the more interesting and mature approaches: the ones addressed by more papers, combining techniques and with better evaluation / validation, representing almost 70% of the publications. These initiatives are:

G04 - LGIL: This research group provided only one paper [#05, 2008], but that is very important for the topic, since it is the first publication proposing a harmonization approach. Actually, Ferchichi and colleagues have published their approach one year before in a borderline paper (FERCHICHI et al., 2007). Thus, ten years ago they proposed a harmonization approach using an ontology as common model and combining the homogenization, mapping and integration techniques in a four steps process. This work influenced some of the following, in particular those from G03 and G10.

G09 - UTS: This research group provided three papers: [#13, 2012] makes a general discussion on standards harmonization efforts and proposes the creation and use of a unification metamodel (for ISO); [#21, 2014] proposes an ontological framework as a reference infrastructure for harmonizing the ISO SC7 standards and some guidelines for deriving integrated ontologies; and [#25, 2016] provides a proof of concept and an application for the proposed framework. The main idea is to produce a hierarchy of ontologies for harmonizing the ISO SC7 standards and use this ontological framework as knowledge reference for creating and updating ISO standards.

G03 - Alarcos: This is the research group with more contributions for our mapping: seven papers. The first [#03, 2006] produces an ontology, using standards and other sources, for harmonizing the software measurement domain. Three papers propose a harmonization framework with a detailed process [#18, 2013], making use of two ontologies [#09, 2012] and [#20, 2014], for managing and executing harmonization initiatives for multi-model environments. Two papers focus on some additional aspects of its framework and in its application in real cases [#10, 2012] and [#24, 2016]. Finally, [#23, 2015] adds new case studies and some detailed data on the integration method. This group published also several other borderline papers, not included in our final selection but that somehow have contributed to their results.

G10 - Aachen: This research group proposes an approach based on metamodels and comparison techniques for mapping and integrating standards [#17, 2013], making use of automated concepts extraction [#14, 2012], similarity metrics [#15, 2012], and automated comparisons [#16, 2013]. These papers are the ones presenting more technical details on how to map and integrate the standards' knowledge by means of common models and semantic relations. This group also published borderline papers not included here.

G06 - NEMO: This research group provided three papers. The first two create ontologies for the software process [#07, 2009] and software measurement [#19, 2013] domains, harmonizing and mapping selected standards. The third one [#22, 2015] proposes software process ontology patterns for deriving integrated domain ontologies from the standards' content.

As it can be seen, some of the groups started the efforts by producing ontologies or metamodels harmonizing the concepts of a target domain ([#13] from G09, [#03] from G03, and [#07] and [#19] from G06) and then migrated to more elaborated harmonization approaches. The resulting approaches share very similar goals, aiming at producing an integrated artifact representing the involved standards. However, the techniques are used with some variations in the order and in the way they are applied. To better understand and compare these initiatives, we have selected the most common steps found by analyzing all the papers, namely:

- **<u>Analyze</u> the standards**, for understanding them, their format and the parts to be used.
- **<u>Extract</u> the information**, by selecting the concepts / definitions / practices to be harmonized.
- **<u>Format</u> the information**, by applying a common structure (homogenization).
- **<u>Model</u> the information**, by modeling the standards' content / structural information.
- **<u>Map</u> the information**, by linking the standards' parts to other standards or a common model.
- **<u>Integrate</u> the information**, by creating an integrated schema representing the standards.
- **Produce a textual <u>Output</u>**, by producing a standard-like text describing the integrated schema.

Matching these steps to the proposed harmonization approaches, we can better understand their behavior and how the steps can be combined. In a nutshell, the selected initiatives organize these steps as shown in Figure 3.8 and described in the following.

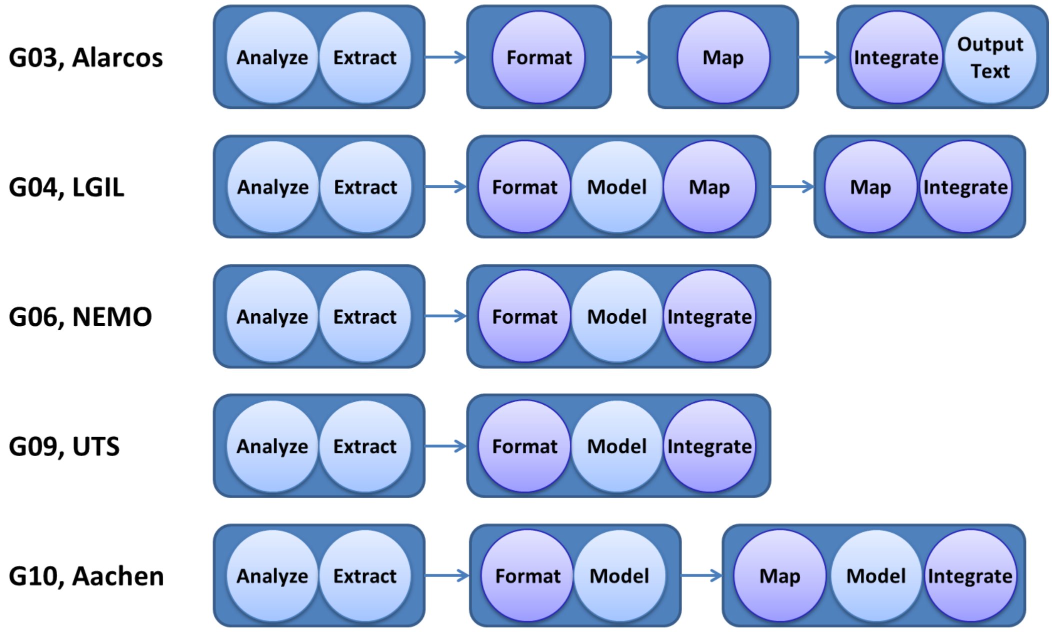


Figure 3.8. Harmonization Approaches Steps.

G03 - Alarcos: Analyzes the standards to extract information (1,2); applies methods to homogenize the information (3), to map them (5), and to produce integrated descriptions (6,7).

G04 - LGIL: Analyzes the standards (1,2); instantiates an ontology that models and maps the information (3,4,5); and produces an integrated schema (4,6).

G06 - NEMO: Analyzes the standards (1,2); and uses their information for deriving a harmonized domain ontology from a core ontology (3,4,6).

G09 - UTS: Analyzes the standards (1,2) and, from a higher-level ontology, instantiates harmonized ontologies (3,4,6).

G10 - Aachen: Analyzes the standards (1,2); models them individually from a metamodel (3,4); and maps them producing an integrated model (5,4,6).

In general, all the mentioned approaches start by analyzing the standards (step 1) and extracting their information (step 2), and produce as result an integrated representation of the information (step 6). However, we have some variation in the intermediary steps, especially concerning modeling. Some initiatives model each standard's information individually and then integrate it (step 4, then 6) (G04, G10); some prefer to produce directly an integrated model (steps 4 and 6) (G06, G09); and one approach does not produce models but a textual standard-like integrated result (steps 6 and 7) (G03). These steps are mostly done focusing on the standards' contents, since the standards' structures usually have been dealt before, as part of the approaches themselves.

Moreover, it is important to say that the presented steps consider the main approach (the most complete and updated one) described by each group, summing up the considered papers. In some cases, different publications of the same group deal with distinct steps before proposing an approach. It is the case of G06 (NEMO) that has not considered the step 5 (mapping) in their main approach presented in [#22], but used it before, in [#07] and [#09]. Actually, even [#22] does not present a complete harmonization approach, but a solution for deriving, from a core ontology (structural model), domain ontologies integrating the involved standards. Similarly, G03 (Alarcos) has not considered the step 4 (modeling) in their HFramework [#18], although they have built an ontology from standards in [#03] and proposed ontologies with structural elements in [#09] and [#20]. Thus, a common behavior in the groups is to have different publications dealing with different techniques, steps or applications. This joint analysis helped us understanding the main steps composing the available harmonization approaches and how they behave in different proposals. It can be useful for improving or defining new solutions on this topic.

Another important issue is how these approaches use the common conceptual models. Group G04 (LGIL) instantiates, from their ontology of structural elements, both the standards' content elements and the mappings between them, from which the integration work is done. G09 (UTS) proposes a high-level ontology with the most general SE notions to be derived to lower levels ontologies according to standards' groups and then specific standards, using these models as reference for creating and updating ISO standards. G06 (NEMO) provides core ontologies and ontology patterns which are used for creating domain ontologies representing an integrated view of the contents of multiple standards on the same domain. G03 (Alarcos) provides a harmonization framework where an ontology with the structure of software processes is used for creating a template (Common Structure of Process Elements) to be applied for homogenizing the involved standards, easing the succeeding mapping and integration efforts. Finally, G10 (Aachen) represents each involved standard in a model instantiated from a structural metamodel, and the elements of these standard's models are linked to an Integration Concept Model built using semantic relations and typed concepts. Analyzing these approaches, it is possible to see a trend for the use of semantic resources, when the models are used as bases for formatting, mapping, integrating or providing domain knowledge. These variant applications of the common models and resulting integrated artifacts are quite interesting and can show some paths for our and future research.

Particularly, the papers using ontologies (63% of the publications in this decade and all in the last three years) point out that they can be useful in diverse senses, such as: (i) representing the standards' concepts and relations between them, including similarity relations [#16]; (ii) providing a vocabulary (including concepts and relationships) specialized to the topic of harmonization [#09]; (iii) eliminating inconsistencies, confusion and terminological conflicts, bringing about benefits such as the provision of precise and clear definitions and a more straightforward representation of processes and standards

[#09]; (iv) being a coherent underpinning for a large group of standards [#21]; (v) providing grounded modeling patterns for representing software process knowledge [#22]; (vi) being used to better understand the standards and their domains and to support the development of tools [#16]. We believe in the potential of ontologies for standards harmonization ends, since it is, in essence, a semantic interoperability problem. Ontologies are a valuable resource to assist each of the mentioned steps by providing domain knowledge, being reference for homogenization and mappings, supporting the modeling and integration of standards' knowledge, easing the link of similarities and the identification of incompatibilities, and providing a consensual representation of the involved standards.

Finally, considering the research types facet, all papers proposed a solution for SE standards harmonization. Half of them were validated by study cases or proper examples. However, only 23% were evaluated in practice, being applied in real word projects (by only two groups, G03 and G04). This shows an interest of the community on assessing the solutions proposed, but a low proximity with the industry, especially considering a topic intending practical benefits for the standards' users and developers. The studies presenting the evaluated approaches reported interesting results on the involved organizations and a relevant feedback to improve the harmonization approaches. Thus, evaluation research should be stimulated in future publications.

3.5 Limitations

It is important to point out the limitations of this mapping study. Concerning the steps for paper selection and data extraction, they were firstly performed only by the doctorate candidate, possibly introducing some subjectivity. For reducing this bias, we defined some objective criteria, added additional reviewers and resolved disagreements when needed, the most used strategies according to Petersen et al. (2015). Thus, the advisors reviewed the decisions made. In the first selection stage (selection by title, abstract and keywords), they analyzed all the selected papers and a sample of 30% of the discarded ones. From the second stage on, all the decisions (selections and classifications) were reviewed by at least one advisor. To reduce bias, the divergent classifications were discussed to reach a consensus.

Regarding the search string, some publications could be missed due to terminological problems. For mitigating these problems, we performed previous simulations in the selected databases and included a number of terms' variations in the search string. We did not search in any particular conference proceedings, journals, or the gray literature (technical reports and works in progress). Only peer-reviewed papers, indexed by the selected electronic databases and those obtained by snowballing and manual search for publications from researchers / research groups were considered. Excluding other sources makes the study more repeatable, but at the risk of missing some relevant paper. Moreover, the additional searches by snowballing and researchers / research groups allowed a larger coverage and more confidence that the papers relevant for the mapping scope have been analyzed and selected. Considering snowballing, from the 58 publications found, 19 were analyzed before (being 9 in our final selection), and only 2 more were selected. Regarding the search by researchers / research groups, from the 95 publications found, 51 were analyzed before (including all the 24 previously selected), and only 2 more were selected. After these additional searches including four more papers to our selection, the perception is that we have selected a representative view of the topic and there are not many publications out of our sight.

Although the ambition of a mapping study is to summarize all the relevant research regarding a topic, different sets of papers can be obtained given the number of decisions taken in secondary studies. The decisions taken by researchers and the judgments exercised affect the results regarding which papers are found and selected and which conclusions the researchers achieve from their secondary studies (WOHLIN et al., 2013). Moreover, several difficulties in performing these studies are pointed out by Wohlin et al. (2013). Some of them we have clearly experienced, such as those related to the criteria for inclusion / exclusion, how the topic and related terms are actually defined, classification schemes, and the influence from the defined research questions.

Regarding the terms used by the researchers to refer the harmonization techniques and the types of common models, we have faced some difficulty due to the lack of uniformity. These terms are not well defined and presented some variations along the publications and time. Concerning the techniques, *homogenization*, *comparison*, *mapping*, *integration*, *unification*, and *harmonization* do not mean exactly the same in the analyzed papers. This fact has affected both the search and the classification of the papers. We tried to mitigate this problem by introducing the most common terms in the search string, with some variations for broadly covering the area. We also used the classification provided by Pardo et al. (2012) to better define when a paper performed comparisons, mappings, integration and so on. Regarding the types of common models, we have also included variant terms in the search string for a wider coverage and decided to classify the models according to the terms adopted by the authors.

Another possible limitation regards the multiple papers produced by the same research group. As the research advanced, the groups published “parts of the story”. Thus, some publications describe only part of a bigger solution, and sometimes this affected the paper selection and the classification of the defined facets. Some publications were subject of debate until we decided by the selection or for a suitable classification. For example, a paper discussing specifically one aspect or technique was classified only for this aspect / technique (e.g., [#20]), even when the whole initiative addresses other aspects / techniques. The level of descriptions also influenced the classifications. Only mentioning an aspect was not enough to classify the paper on that aspect, it needs some discussion / demonstration, even when it is included in a broader initiative. Moreover, sometimes a given paper presents a relevant solution also discussed in another already considered paper, plus a contribution that is not in the scope of our mapping (e.g. a supporting tool (PARDO et al., 2011), or a case study (PARDO et al., 2016b)). Even passing in our inclusion criteria, IC1, we discarded such publication by EC5 (*The paper is an older version of other paper already considered*). On the opposite way, some papers presenting solutions already discussed in other considered paper but adding a new contribution or further discussion for the mapping scope were considered (e.g., [#23, #24]). These more detailed criteria were discussed and decided, by all the authors, after selecting the mentioned publications and analyzing them by groups.

Lastly, there is also much judgment involved in the classification schemes and in the decisions taken to fit the papers in the proper categories. Concerning the specific categories, as aforementioned, we needed several discussions before finding a more precise classification for some papers regarding harmonization perspectives and techniques, and the types of common models. Referring to the general categories, we have used the classification scheme for research type proposed by Wieringa et al. (2006) (originally proposed for use in requirements engineering) and updated by Petersen et al. (2015), where it is pointed as a frequently used classification in mapping studies. Furthermore, classification is a judgment influenced by the researchers' background and expertise. Thus, some subjectivity might be introduced in the classification actions.

3.6 Related Work

During this research, we found a secondary study on the same topic, with a similar scope. Pardo and colleagues (2010) have published a systematic review on Harmonization of Reference Models[5], with the aim of "analyzing the state of the art with regard to initiatives concerning the harmonization of multiple reference models". They selected 32 papers and, as a systematic review, conducted a deeper analysis focusing on how the harmonization initiatives work, which are the standards used, and the methods and techniques proposed. Finally, they propose a harmonization framework which is considered in our mapping, being described in the papers from the group G03 (Alarcos).

By comparing our mapping to Pardo's study (PARDO et al., 2010), some differences are identified. Firstly, although both studies address the same topic, they have an important distinction in the focus. While this mapping searched only for standards harmonization papers *using a common conceptual model*, Pardo's study sought works harmonizing or integrating reference models (presenting a technique for that or doing it in practice). As a consequence, only we selected papers such as [#02, #03 and #07], which provide an ontology mapped to standards, but without describing any harmonization technique; while only Pardo et al. selected papers directly harmonizing standards without a general common model, such as (PAULK, 1993) (YOO et al., 2006) (ROUT; TUFFLEY, 2007). Secondly, we

[5] Pardo and colleagues (2010) apply to "Reference Model" the same meaning we are applying to "Standards" in this work.

have used different databases as sources. Pardo et al. did not use *Scopus*, *Compendex* and *Web of Science*, while we have not used *Wiley InterScience* and gray literature. Finally, they conducted the searches in 2010, while our mapping includes publications until 2016, which makes a significant difference in this case. As discussed, most of the papers we have selected are from 2011 on (19 out of 26), including even relevant contributions of Pardo and colleagues, produced after their review.

For a more detailed comparison on what papers were selected, considering the period in common (until 2010), Pardo et al. have selected 32 papers while this mapping has selected 7. However due to the differences regarding the types of publication and the focus given, only one paper was finally selected by both. Pardo et al. have not considered papers presenting ontologies or models for harmonization if they are not used in practice for that (such as their own group's paper [#03] and other [#02 and #07]). Analyzing the 32 Pardo's selected publications, 13 are technical reports or books that we have not considered for not being peer reviewed. From the other 19, according to Pardo's classification, 11 presented a direct comparison between specific standards, without the use of a common model. The 8 remaining were analyzed in our selection process, being 7 discarded for not satisfying IC1 (*The paper addresses harmonization of SE standards by means of a common model*) and only one finally selected, the Ferchichi's paper [#05]. Moreover, from the 19 peer reviewed papers, we analyzed 15 in the selection process and checked the other 4 to avoid missing papers.

In spite of these differences, there are certain similarities that helped us define some parameters and to confirm some trends. Since they made a deeper analysis on the harmonization techniques, we borrowed some terms to improve our search string (see Table 3.2) and some definitions to better define our techniques classification (classification scheme for RQ3). Their search string has 3 groups of terms considering *approach*, *subject* and *area*. We have used the very same terms for the *approach*'s group, only adding "interoperability" and "interoperable". For the *area*'s group, besides their terms "software process" and "process improvement", we have included "software engineering" and "SPI". For the *subject*'s group, we have not used the terms "models", "frameworks" and "technologies" due to the high volume of results, with little difference in what we are searching for. Finally, we have included a new group *means*, for dealing with the common models.

Regarding the studies' results, Pardo et al. pointed out a growth in the interest for the standards harmonization topic becoming more evident from the second half of the 2000's. Our results corroborate the increase in interest for standards harmonization (using a common model, in our case) a little later, being accentuated in the first half of the current decade. Figure 3.9 shows the number of selected papers in each study organized by lustrums (five years). Moreover, we also checked Pardo et al. work for confirming the most used standards and techniques.

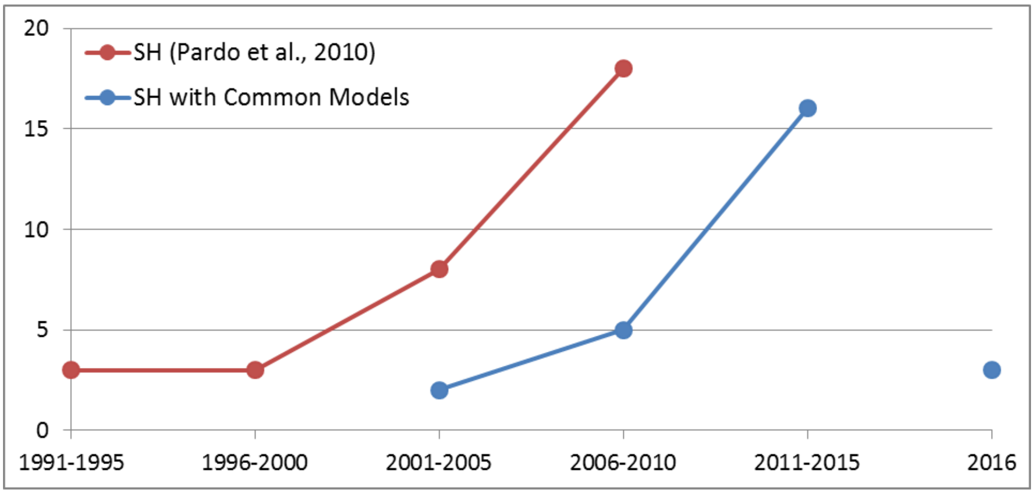


Figure 3.9. Trends on Standards Harmonization.

Finally, Pardo and colleagues concluded that "there is currently a lack of guidelines with which to help organizations to implement the harmonization of multiple models, and of a unified terminology with which to homogenize the diversity of the structure of the different models and the harmonization techniques which can be applied" (PARDO et al., 2010). Based on our study, we do agree with these conclusions. We have evidenced that the approaches and techniques for standards harmonization are evolving, and more and more publications are using common models as reference for unifying terminology and semantics to support the harmonization efforts.

The papers we have analyzed provided valuable contributions to reduce this gap. Taking a broader view on Pardo's work (within the Alarcos group), they proposed a harmonization framework able to support the definition and execution of a harmonization strategy. The framework includes a process with a set of guidelines organized as methods for homogenization, mapping and integration; and a pair of ontologies providing a unified terminology for the harmonization realm, and for representing the structural elements of the standards' processes. Each standard is homogenized when represented in a template defined from the process ontology and then they can be mapped and integrated according to the defined methods. A web tool supports the process. Moreover, they have applied the framework in several harmonization projects (including industry). In turn, the research in the other publications also advanced, as discussed before.

However, we believe there is still a research gap to be dealt with. Regarding the conceptual aspects, more than a unified terminology, we still need a semantic referential representing both structural and content standards' elements. Like most of the recent publications, we also trust on the potential of ontologies, but they shall be reliable and comprehensive enough to provide an unambiguous representation of the domain knowledge related to the standards being harmonized. This ontological referential should be able to connect the similarities and offer a semantic endowed solution for the divergences while harmonizing standards. Regarding the methodological aspects, the guidelines and methods must be highly aligned to the ontologies, exploring their applications. Such methods should allow an effective semantic judgment, better classifying the standards' elements, dealing with the variant types of relations between them, and offering the semantic and structural basis for an integrated representation of the involved standards. We see standards harmonization as a semantic interoperability problem, claiming for a semantic solution.

Moreover, we believe this is a wider issue, affecting not only the organizations that adopt the standards, but everyone who produce, study or apply them.

3.7 Final Considerations

This chapter presented a systematic mapping on SE Standards Harmonization using a Common Conceptual Model. Eight research questions were investigated concerning the following facets: (i) distribution of the selected papers over the years, venues and groups; (ii) harmonization perspectives applied; (iii) harmonization techniques applied; (iv) harmonization subject areas; (v) standards being harmonized; (vi) types of common models used; (vii) harmonization results focus; and (viii) research types.

Standards Harmonization in SE has shown to be a promising research area, since standards' users and developers frequently need to deal with multiple standards. As previously discussed, the publications on this topic are increasing in the recent years. The main contribution of our mapping is to make more evident some aspects regarding SE standards harmonization by means of common models, to support the development of our research as well as to drive future research in this topic. In this context, the mapping conclusions are summarized as follows: (i) there is increasing interest on the topic in the last five years by the SE research community and also by standardization organizations, such as ISO; (ii) the main area subject of the harmonization initiatives is Software Process, being CMMI the most studied standard; (iii) harmonization techniques are evolving, using more resources and being more frequently combined into approaches; (iv) ontologies are being more adopted as common models to support harmonization efforts; and (v) the researches have not yet reached industry satisfactorily.

Considering the mapping conclusions, a research gap to be explored relates to the harmonization techniques and the application of ontologies. We believe the next step towards standards harmonization clearly is to better deal with semantics. The evolution of harmonization techniques (iii) has been supported by a variety of resources. Considering the existence of a common conceptual model, *ontologies* have been the most preferred model type in the recent years (iv). However, sometimes, "ontology" is just a beautiful name for a conceptual model, other times there is a real semantic advantage on applying it. The uses vary, being a simple generic model to be instantiated, a knowledge source for supporting the harmonization techniques, or a knowledge model able to derive harmonized representations. In general, the represented information covers a narrow portion of the SE domain, mainly focusing on structural aspects, barely exploring the potential of ontologies in this topic.

We believe that ontologies can be created to be a broader knowledge framework, able to represent even specific domains in SE. To be taken as a reference, they should be well-founded and provide a reliable representation of reality. Foundational ontologies are a useful resource for this purpose. Among the groups addressing standards harmonization, some discuss the importance of using higher-level ontologies for grounding the lower level models. For instance, in G09, Henderson-Sellers et al. (2014) considers extending their proposed ontological framework by including, in the future, an Advanced Foundational Ontology for Standards (AFOS). However, we think that a foundational ontology must be the basis, providing the essential distinctions and enriching the semantics of the ontologies used in standards harmonization. Using well-founded SE ontologies would make it easier to deal with standards' similarities and divergences, to support harmonization techniques, and thus to achieve a more suitable solution for exploring the potential of ontologies for harmonization efforts.

Aiming at reaching this goal, we have worked on two related fronts: (i) building SEON, a well-founded Software Engineering Ontology Network for representing general and specific SE knowledge for better supporting harmonization initiatives; and, (ii) developing *Harmony*, supported by the best practices and findings identified in this mapping, this harmonization approach shall combine the proper techniques and explore SEON capabilities and semantic resources for better dealing with semantic aspects of standards harmonization. SEON is presented in the next chapter and Harmony is addressed in Chapter 5.

4 A Software Engineering Ontology Network

This chapter presents the Software Engineering Ontology Network (SEON), the knowledge framework used for supporting the harmonization efforts in this research. It presents SEON's requirements and architecture; how the core layer is built to support incorporating new domain ontologies and to favor semantic integration / harmonization initiatives; and the current domain networked ontologies and integration mechanisms. Finally, it discusses related works and conclusions.

4.1 Introduction

Ontologies have been widely recognized as a key enabling technology in a variety of initiatives such as Knowledge Management (KM) (O'LEARY, 1998), Semantic Web (SUÁREZ-FIGUEROA et al., 2012), Tool and System Integration (CALHAU; FALBO, 2010) (FALBO et al., 2005), Semantic Documentation (ERIKSSON, 2007) and Standards Harmonization (PARDO et al., 2012) (HENDERSON-SELLERS et al., 2014), among others. In KM, ontologies are used for establishing a common conceptualization of the domain of interest to support knowledge representation, integration, storage, search and communication (O'LEARY, 1998). The Semantic Web is characterized by the existence of a very large number of distributed semantic resources, which subscribe to alternative but often overlapping modeling schema (i.e., ontologies) (SUÁREZ-FIGUEROA et al., 2012). In Standards Harmonization, ontologies are useful for providing a referential knowledge in the subject domain favoring the comparison and integration of related standards (PARDO et al., 2012). These initiatives can also be related by common aspects addressing the need for semantic solutions and the integration of knowledge resources.

However, sometimes, just selecting or building an ontology and using it is not enough. This occurs especially in applications dealing with large and complex domains, requiring a broader solution. It is the case of Software Engineering (SE), our application domain. If we try to represent the whole domain as a single ontology, we will achieve a large and monolithic ontology that is hard to manipulate, use, and maintain (SUÁREZ-FIGUEROA et al., 2012). On the other hand, representing each subdomain in isolation would be too costly, fragmented, and, again, hard to handle (RUY et al., 2016).

A variety of ontologies has been developed for modeling the SE domain, attempting both approaches. According to Calero et al. (2006), these ontologies can be classified as: Generic SE Ontologies, having the ambitious goal of modeling the complete SE body of knowledge; or Specific SE Ontologies, attempting to conceptualize only part (a subdomain) of this discipline.

Concerning Generic SE Ontologies, Mendes and Abran (2005) propose a SE ontology consisting of an almost literal transcription of the SWEBOK (BOURQUE; FAIRLEY, 2014) text, with over 4,000 concepts. Sicilia and colleagues (2005) propose an ontology structure to characterize artifacts and activities, also based on SWEBOK. Wongthongtham and colleagues (2009) propose an ontology model for representing the SE knowledge, based on SWEBOK and Sommerville's Software Engineering book (SOMMERVILLE, 2004). These ontologies, although grounded on acknowledged references, use just a single or a couple of them, disregarding a relevant ontological principle: to provide a shared conceptualization (GRUBER, 1995).

Considering the Specific SE Ontologies, a great number of them is available, representing a variety of SE subdomains, such as Software (e.g., (OBERLE et al., 2009) (MALONE et al., 2014)), Software Process (e.g., (GONZALEZ-PEREZ; HENDERSON-SELLERS, 2006) (BRINGUENTE et al., 2011)), Software Requirements (e.g., (WANG et al., 2014)), Software Testing (e.g. (SOUZA et al., 2013)), Software Configuration Management (e.g., (CALHAU; FALBO, 2010)), among others (CALERO et al., 2006). These subdomain ontologies are frequently not or weakly interrelated, and they are often built and applied in isolation.

However, let's consider that the subdomain ontologies are developed bearing in mind their integration with others. Taking this to the extreme, the combination of ontologies of all SE subdomains would result in an ontology of the complete SE domain (CALERO et al., 2006). Unfortunately, the reality is that this goal is extremely laborious. Not only due to the size and complexity of the domain, but also due to the numerous problems related to ontology integration and merging (CALERO et al., 2006), such as overlapping concepts, distinct foundational theories, and different representation and description levels, among others. Moreover, like in other domains, SE subdomains share concepts, ranging from general (e.g. *Artifact*, *Process*) to specific ones (e.g. *Functional Requirement*, *Test Case*, and *Nonconformity*). This important feature of the SE domain must be considered while representing it.

For achieving consistent SE ontologies, concepts and relations should keep the same meaning in any related ontology, and meaning assignment cannot be done in a bottom-up way. In sum, SE comprises a set of highly interconnected subdomains sharing a number of concepts and definitions. This interrelated nature affects any possible representation of the SE domain, and the situations in which it can be applied. Despite of the challenges involved, an ontological representation covering a large extension of the SE domain remains as a desired solution.

D'Aquin and Gangemi (2011) point out a set of characteristics that are presented in "beautiful ontologies" (i.e. good quality ontologies), from which we detach the following ones: having a good domain coverage; being modular or embedded in a modular framework; being formally rigorous; capturing also non-taxonomic relations; and reusing foundational ontologies. Most of the existing SE ontologies do not exhibit such characteristics. We believe that an integrated ontological framework, built considering them, can improve ontology-based applications in diverse initiatives addressing the SE area. In such integrated framework, there must be ways for creating, integrating and evolving related ontologies. Thus, we advocate that these ontologies should be built incrementally and in an integrated way, as a *network*. Taking these considerations in mind, we propose **SEON**, a **Software Engineering Ontology Network**.

This chapter discusses how SEON is being developed. Section 4.2 presents the main requirements that guided the network design. The SEON Architecture and how it was designed for assuring the necessary grounding for the networked ontologies is presented in Section 4.3. Section 4.4 discusses the SEON core layer, presenting the current version of the Software Process Ontology and how it has evolved for being aligned to general SE standards and for being organized as an ontology pattern language, easing the network growth. Section 4.5 discusses how domain networked ontologies can be created, integrated and evolved, showing as an example the Quality Assurance Process Ontology. Section 4.6 discusses how the SEON Specification favors the network access and use. Section 4.7 presents related works, and Section 4.8 our final considerations.

4.2 SEON Requirements

SEON aims at providing a knowledge framework able to represent an increasing portion of the SE domain, to support semantic interoperability initiatives in general, and those addressing SE standards harmonization in particular.

Although SEON itself is a new proposal, the studies and ontologies developed by our group along the years were essential contributions for defining the network. Diverse efforts have been made for specifying, designing and building specific SE ontologies (e.g. (DUARTE; FALBO, 2000) (NARDI; FALBO, 2006) (CALHAU; FALBO, 2012) (SOUZA et al., 2013)), reusing knowledge from higher level ontologies (e.g. (GUIZZARDI et al., 2008) (BARCELLOS et al., 2010)), restructuring ontologies (e.g. (FALBO; BERTOLLO, 2009) (BRINGUENTE et al., 2011)), ontologically analyzing conceptual models (e.g. (ALMEIDA et al., 2010) (RUY et al., 2014)), defining and applying ontology patterns (e.g. (FALBO et al., 2013) (FALBO et al., 2016) (RUY et al., 2017)), integrating ontologies (e.g. (RUY et al., 2016)), using ontologies for applications demanding integration (e.g. (FALBO et al., 2005) (CALHAU; FALBO, 2010)) and so on. These efforts have, along the time, provided us the knowledge and elements necessary for a research step forward, supporting an essential purpose of this thesis: the SE Ontology Network.

For designing SEON, as an ontology network (ON), several decisions have been taken to better define its essential characteristics. These requirements were based on diverse sources. Suárez-Figueroa and colleagues (2012) provided the main notions about ONs; the research gaps (discussed in Chapter 3) revealed the need for a comprehensive and consistent SE domain representation; our group's experiences and previous works offered a rich material and diverse ideas on how to organize and complement it; and finally, the own network development and application afforded us with valuable findings for improving SEON. This section presents the characteristics we judged essential for guiding SEON development and future improvements. In the following, we discuss why these requirements were chosen and what we expect by considering them.

Ontology Network. *The SE knowledge framework shall be structured as an Ontology Network.* ONs are able to organize a set of interrelated ontologies seeking for important aspects such as modularity, consistence and reuse. This is crucial feature for representing a broad knowledge area with diverse subdivisions sharing a set of common concepts, such as the SE domain. Moreover, ONs are more suitable for accommodating changes and incrementally growing as it is expected in SEON.

Generality Levels. *The networked ontologies composing SEON shall be organized according to generality levels.* Besides the diverse subdomains to be represented, the ontologies can also vary in abstraction, presenting different generality levels. An ON can take advantage of this to encourage reuse from the higher to the lower generality levels. This way, *general knowledge* (from a foundational ontology) can be reused for describing *general SE knowledge valid for several subdomains* (in a core ontology), that, in turn, can be reused for representing *subdomain specific knowledge* (domain ontologies). The ontology generality levels directly affect the ON architecture and reuse mechanisms.

Foundational Ontology. *A foundational ontology shall provide the (same) foundation for all networked ontologies.* The use of foundational ontologies is a recommended practice in Ontology Engineering (D'AQUIN; GANGEMI, 2011) due to its capacity to better describe reality (GUIZZARDI, 2005). In an ON, a foundational ontology can be adopted as reference general knowledge, providing a common grounding for all networked ontologies. It is essential for assuring compatibility between the ontology representations, thus supporting integration efforts. A foundational ontology is also useful for conducting ontological analysis on external knowledge sources, in particular existing SE subdomain ontologies that we want to integrate to SEON.

Reference Ontologies. *The networked ontologies shall be reference ontologies, prioritizing human usage.* One of the SEON's main goals is providing a reliable and consensual representation of the SE domain for supporting semantic-related problems. Standards harmonization, like other integration initiatives, is predominately a human activity, requiring understanding, analyzing and integrating different sources of knowledge, using a semantic referential. Thus, SEON models shall be developed prioritizing human understanding and usage.

Reuse Mechanisms. *Core ontologies shall be organized for favoring reuse.* Most of SEON's content is given by its domain ontologies. Following the principle that the higher-level ontologies should support the development of the lower level ontologies, it is expected that SEON provides mechanisms for reusing core ontologies in the development of domain ontologies. Ontology Patterns is a promising approach for reusing SE general knowledge in the domain ontologies. It improves the development support and productivity, and makes the domain ontologies more consistent (RUY et al., 2015b). Besides that, ontology patterns favor an analogous representation for similar domain fragments, which is important for the network integration. In particular, Domain-Related Ontology Patterns (DROPs) generally are strongly related, so that a pattern only makes sense to the extent that it is supported by other patterns. This occurs because the features introduced by applying one pattern may be required by the next. In such situations, a larger context is needed to describe the larger problems that can be solved by combining patterns, and to address issues that arise when patterns are used in combination. In such cases, we need to put the patterns together in an Ontology Pattern Language (OPL) (FALBO et al., 2013). Thus, for capturing the most general SE knowledge in SEON, we advocate in favor of organizing the SEON core ontologies as OPLs.

Network Evolution*. The network shall provide mechanisms for favoring its own evolution.* SEON is not a complete network. The SE knowledge is very wide and SEON currently represents only a small portion of it. However, for keeping the network growing, we must provide the proper mechanisms for favoring its evolution. For example, a well-defined architecture can organize the existing ontologies and accommodate the addition of the new ones; the foundational grounding helps to make the ontologies more consistent; reuse mechanisms favor the creation and integration of new ontologies; integration mechanisms help incorporating new knowledge sources; and the way SEON is represented and made available for potential users eases its access and application.

Once the SEON requirements are presented, it is important to highlight that they do not constitute all the requirements for the network. However, they are the most important ones, taken as the basis for designing, developing and improving SEON with the characteristics we have considered essential.

4.3 SEON Architecture

SEON is founded on a number of experiences, ideas and findings on ontology development and application, especially in the SE field. As mentioned before, for building such integrated framework, it is not enough to put the pieces together. It is an incremental and long-term work. Ontology development and integration are not simple tasks. Thus, to truly enjoy the benefits of keeping the ontologies in a network, we need to take advantage of the existing resources available in the ON for gradually improving and extending it. In other words, the ON shall provide the means for facilitating its consistent growth. The ON must provide the high-level structures and methods for favoring the accommodation of the new, lower-level, pieces. Networked ontologies composing an ON shall be incorporated in such a way that preserves the network properties and does not fall in conflict with any existing part. Finally, it is not our ambition to establish a “complete SE ontology”, but to provide a starting point, and a proper structure and features for easing the addition of new ontologies and evolution of the existing ones.

SEON development starts by the definition of its architecture. In this context, three main SEON premises, based on the network requirements, are important for defining our architectural solution:

(i) **Grounding**: being based on a well-founded grounding for supporting ontology development;

(ii) **Growth**: offering mechanisms to support building and incorporating new SE subdomain ontologies to the network; and

(iii) **Consistency**: promoting integration by keeping a consistent semantics for concepts and relations along the whole network.

Taking the above premises, an ON should be organized in a *layered architecture*. In the background, we need a foundational ontology to provide the general ground knowledge for classifying concepts and relations in the ON. In the center of the ON, core ontologies should be used to represent the general domain knowledge (on SE), being the basis for the domain ontologies. Ideally, these core ontologies should be organized as Ontology Pattern Languages (OPLs) to favor reusing model fragments (ontology patterns) while developing the domain ontologies. Finally, going to the extremities, (sub) domain ontologies appear, describing the more specific knowledge.

Therefore, the SEON architecture is organized considering the stated premises and three main ontology generality levels, as Figure 4.1 shows.

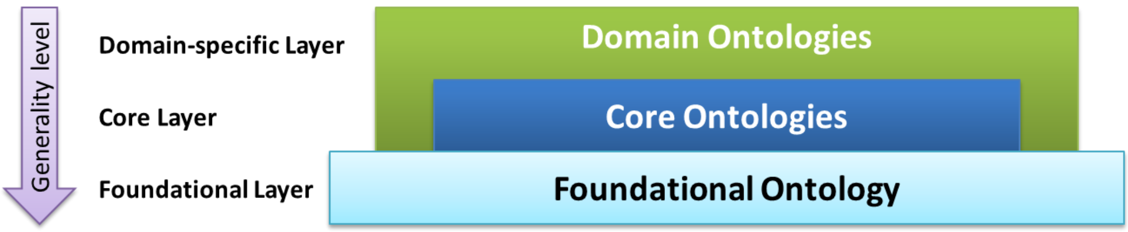


Figure 4.1. The SEON Architecture.

Foundational Layer

At the bottom of SEON, sustaining the network, is the Unified Foundational Ontology (UFO). UFO is divided in three parts: UFO-A, an ontology of endurants (objects) (GUIZZARDI, 2005), UFO-B, an ontology of perdurants (events) (GUIZZARDI et al., 2013), and UFO-C, an ontology of social entities (GUIZZARDI et al., 2008). UFO-C builds on top of UFO-A and UFO-B. UFO ontological distinctions are used for classifying the SEON concepts, e.g., as *objects*, *actions*, *commitments*, *agents*, *roles*, *goals* and so on. UFO provides the necessary grounding for the concepts and relations of all networked ontologies. Although UFO is incorporated as an essential part of the ON, it is not under SEON control, being evolved apart. It is presented as a background of this research in Section 2.3, and used in diverse situations henceforth.

Core Layer

In the center of SEON, providing the SE core knowledge for the network, there is the Software Process Ontology (SPO) (BRINGUENTE et al., 2011). SPO is a core ontology grounded in UFO, aiming at establishing a common conceptualization on software processes. SPO scope embraces the following aspects of the software process domain: standard, intended and performed processes, their activities; artifacts handled, resources used and procedures adopted, and stakeholders' participation. SPO reuses some concepts from the Enterprise Ontology (EO), a core ontology on enterprises (FALBO et al., 2014) (which we consider external to SEON), for dealing with aspects related to organizations, such as team membership. Another external core ontology related to this layer is the Core Ontology on Measurements (COM) (BARCELLOS et al., 2014). All these core ontologies are organized as Ontology Pattern Languages (OPLs). The Core Layer and SPO are further discussed in Section 4.4. Differently from the foundational layer, which admits only a single foundational ontology (for providing the same theory to the whole network), the core layer can support multiple, compatible, core ontologies (e.g. core ontologies describing Artifacts, Resources or Software).

Domain-specific Layer

Over the foundational and core layers, domain ontologies appear. Each domain networked ontology is grounded in the core ontologies and also in UFO, and encompasses a portion of the SE domain (e.g., software requirements, design, configuration management, quality assurance). Although not explicit in Figure 4.1, according to the generality levels continuum (see Section 2.2), more specific domain ontologies can be developed based on other more general domain ontologies. For instance, ontologies addressing runtime requirements (RRO) (DUARTE et al., 2016), goal-oriented requirements (GORO) (NEGRI et al., 2017) and the requirements development process (RDPO) were developed based on the Reference Software Requirements Ontology (RSRO). The Domain-specific Layer and its domain ontologies are discussed in Section 4.5. For matter of organization, domain networked ontologies can be grouped in subnetworks, as the Requirements engineering Ontology subNetwork (ReqON), which comprises the four aforementioned requirements related domain ontologies.

In a nutshell, the foundational layer offers the ontological distinctions for the core and domain layers, while the core layer offers the SE core knowledge for building the domain ontologies. This way of grounding the ontologies in the network is helpful for engineering the networked ontologies, since it provides ontological consistency and makes a number of modeling decisions easier. It supports the **grounding** premise.

The SEON building mechanisms also take advantage of ontology patterns by representing its core layer as OPLs. SPO, as well as the other core ontologies, EO and COM, are organized as OPLs to become more modular, flexible and reusable. Thus, the ontology engineer can explore alternative models in the design of specific ontologies for the various SE subdomains, select the ontology fragments relevant to the problem in hands and reuse them (FALBO et al., 2013). During domain ontology development, the ontology engineer selects the suitable patterns and extends their concepts and relations in the networked ontology (Section 4.5 shows a case for the Quality Assurance subdomain). In the cases where a domain element is not covered by the core ontologies, this domain-specific element should be grounded directly in the foundational ontology (UFO). Moreover, SE is a very interrelated domain and, as the size of the ON increases, it has more ontologies with concepts and relations potentially reusable by the new ontologies. This reuse-based development reinforces the **growth** premise.

By reusing patterns (from the core layer), the development of the domain ontologies becomes faster and the resulting models more consistent and uniform (RUY et al., 2015c). The core ontologies, sustained by the foundational ontology, offer a standardized way for describing all the other elements in the network. Thus, since all the domain networked ontologies inherit the same core and foundational grounds, concepts and relations with the same classification have a common and identifiable background. This is a fundamental aspect for ontology integration.

As ontologies are developed and added to SEON, we still need to work on integrating them. Although the domain ontologies share the same conceptual basis, given by the foundational and core ontologies, they still need to be aligned with respect to their specific knowledge. It makes it possible for networked ontologies to be merged in a meaningful way, by representing information in one ontology in terms of the entities in others (SUÁREZ-FIGUEROA et al., 2012). SEON adopts some alignment guidelines for matching and integrating domain networked ontologies (explained in Section 4.5). The integration mechanisms, allied to the shared basis, contribute to the **consistency** premise.

Figure 4.2 shows the current status of SEON. Each circle represents an ontology. The circles' sizes vary according to the ontologies' sizes (in terms of number of concepts, represented inside the circles in parenthesis). Arrows denote dependencies between networked ontologies, and line thickness represents the coupling level between them (in terms of number of relationships between concepts in different ontologies). Blue circles are the core ontologies; and green circles, the domain ontologies. The upper-left area comprising four requirements domain ontologies represents the ReqON subnetwork. The core ontologies EO and COM, although used for modeling SEON domain ontologies are external to the network and thus not represented in this figure.

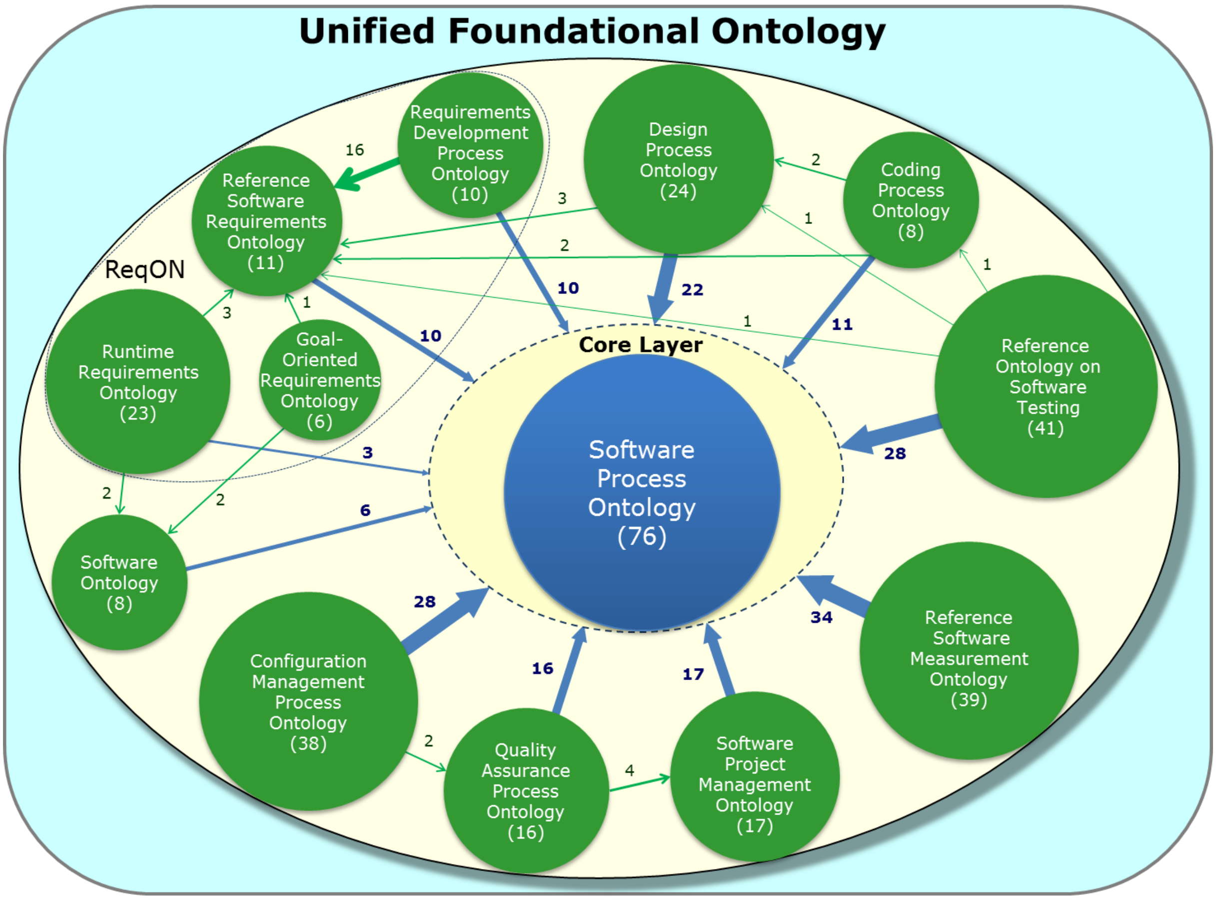


Figure 4.2. SEON: The Network View.

It is important to notice that, even adopting a layered architecture, SEON is a network. Like so, each new added node contributes to the whole network. When a new ontology is added, it should reuse existing elements (from a higher or the same layer). Other ontologies, in turn, may be adapted to keep consistency, in order to share the same semantics along the whole network. Even the core ontologies can evolve to adapt or incorporate new concepts, relations or patterns discovered when domain ontologies are created or integrated. Moreover, due to the size and complexity of the SE domain, it requires a continuous and long-term effort with ontologies being added and integrated incrementally. The next sections further discuss how we have achieved the current status of SEON and how we plan to go ahead.

4.4 Building the SEON Core Layer

The SEON Core Layer is the network's heart, containing the general SE knowledge, common to most domain networked ontologies. As defined in the architecture, SEON core ontologies are grounded in UFO to be built on the same ontological theory, using the very same ontological distinctions. The Core Layer should reach a good coverage on the SE domain to provide a sound support for building any domain ontology. As mentioned, SPO is, currently, the only SEON core ontology. Other core ontologies are also used by SEON, namely as EO and COM, but they are considered external to it, since they are not devoted to the SE domain, being more widely applied.

The software process domain was chosen as main basis of the SEON Core Layer since it describes most of the general notions necessary for representing the SE area, such as processes, activities, artifacts, software and hardware resources, stakeholders, procedures and so on. Thus, since a relevant part of the SEON history is attached to the last evolutions of the Software Process Ontology, it deserves especial attention and a further discussion in this section.

As discussed in Chapter 2, SPO was subject of several evolutions, culminating in its reengineering for being grounded in UFO (BRINGUENTE et al., 2011) and its reorganization as an OPL (SP-OPL) (FALBO et al., 2013). From that point on, the research in the context of this thesis also has been contributing to the SPO evolution in some ways. Actually, most of these improvements are being made with the aim of developing a better product for supporting the harmonization efforts (SEON) of which SPO is an essential part.

Firstly, we conducted an Ontological Analysis of the ISO/IEC 24744 Metamodel: the Software Engineering Metamodel for Development Methodologies (SEMDM) (ISO/IEC, 2007). SEMDM describes general ISO concepts and relations and is intended to be used as basis for the ontological framework proposed by the ISO Harmonization Initiative (HENDERSON-SELLERS et al., 2014) (GONZALEZ-PEREZ et al., 2016). However, we found some conceptual problems in SEMDM and performed an ontological analysis in the light of UFO for proposing solutions in some model fragments (RUY et al., 2014). Since SEMDM describes software processes, much of the SPO rationale applies. Thus, SPO has contributed to the proposed solutions, and some insights were used as feedback for improving SPO. Moreover, since SEMDM uses an ISO standards' vocabulary, this experience was also useful for better aligning some terms and definitions of SPO, and hence SEON, to this realm. The complete SEMDM ontological analysis is presented in (RUY et al., 2014). This study concludes that, for building an ontological framework (in the case, for standards harmonization purposes), the grounding on a foundational ontology is a critical aspect. Regarding SEON, it has reinforced our plans of building the network grounded in UFO.

In another study, we took the fragments proposed in the ontological analysis, represented them as ontology patterns and organized them in an ontology pattern language, called ISO-based Software Process OPL (ISP-OPL). We have also conducted an experiment producing eight domain ontologies from SE standards supported by ISP-OPL (RUY et al., 2015b). This work with ontology patterns showed us diverse improvements applicable for SP-OPL and for our approach to ontology development. This study was a fundamental research step for SEON development. Among the main findings, it is important to highlight that (RUY et al., 2015b): (i) ontologies created with the same grounding, and from the same patterns, have a higher similarity degree, being easier to be reused and integrated; (ii) reuse of higher-level ontologies favors not only building the models, but also the definition of the related competency questions and axioms; and (iii) the use of ontology patterns can make the ontology development easier, more productive and presenting higher quality results. From these findings, we decided to adopt a pattern-based development approach for building SEON domain networked ontologies.

As a side effect of this work applying ISP-OPL, our interest in ontology patterns and OPLs also increased, leading us to improve our methods for creating and applying ontology patterns. As a result, we have proposed an approach for deriving conceptual ontology patterns from foundational and core ontologies, and guidelines for applying them in combination for building domain ontologies in a reused-oriented process (RUY et al., 2017).

Subsection 4.4.1 presents the approach for deriving patterns from core ontologies, which is one of the basis for building the core layer of SEON. Subsection 4.4.2 presents the current SEON Core Layer, introducing its core ontology. Some patterns of SP-OPL are presented in Subsection 4.4.3.

4.4.1 From Core Ontologies to Ontology Patterns

Being ontology patterns one of the main instruments for creating domain ontologies in SEON, it is important to provide the means to develop them. In the proposed layered architecture, the higher-level ontologies are the basis for producing the lower-level ones. UFO, the highest level, is already established. Thus, from the Core Layer on, everything has to be built, including the domain ontologies and also the patterns used for building them. For achieving SEON's goals, as important as developing the ontologies *with reuse* is developing them *for reuse*.

In (RUY et al., 2017), we present an approach for extracting Domain-Related Ontology Patterns (DROPs) from core ontologies, and guidelines describing how ontology patterns can be applied in combination for building reference domain ontologies in a reuse-oriented process. In this section, we focus on the approach for extracting DROPs, which we are using for organizing the SEON core ontologies as OPLs.

Core ontologies are important sources of DROPs, since they describe the core knowledge of a wide domain that spans across different subdomains. Their models contain fragments of knowledge that can be reused when modeling more specific domain ontologies. The main issues related to extracting DROPs from core ontologies are: (i) how to cut the core ontologies, properly defining the fragments boundaries for DROPs, and (ii) how to relate the obtained DROPs. Information from the core ontology can be used for supporting the fragmentation process for creating DROPs. In particular, competency questions (CQs) can reveal modeling needs in small (and still connected) pieces. Next, we present a general process to fragment a core ontology in DROPs, comprising four steps and providing some guidelines.

A. Modularize the Core Ontology.

If the core ontology to be fragmented in DROPs is a complex one, we should start by splitting it into modules (sub-ontologies), for example, using an approach of Ontology Partitioning (D'AQUIN, 2012). As pointed out by d'Aquin (2012), there is no universal way to modularize an ontology and the choice of a particular approach should be guided by the ontology requirements. The general approach proposed in (D'AQUIN, 2012) can be applied for this purpose. The same can be said for techniques and criteria for ontology partitioning, such as the ones proposed in (D'AQUIN et al., 2009). Regarding the criteria, we suggest considering at least the following: independence, cohesion, and size. If the core ontology is already well modularized, this step can be skipped.

B. Fragment Each Sub-Ontology Model into Small Pieces Still Meaningful for the Domain.

Consider splitting a sub-ontology into even smaller fragments suitable to be considered DROPs. By looking again at ontology requirements, we can look at each sub-ontology and search for lower-granularity reusable fragments. A pattern, in general, describes a particular recurring problem that arises in specific contexts and presents a well-proven solution for the problem (BUSCHMANN et al., 2007). Thus, the process of extracting DROPs from a sub-ontology should start by looking for the problems being addressed by fragments of the sub-ontology. If the core ontology is guided by the competency questions (CQs) that it aims at answering, then these CQs are the natural guide for driving the process of splitting the sub-ontology into DROPs. As a starting point, we suggest defining one DROP for each CQ. However, it is important to keep in mind that CQs and DROPs have different concerns. While CQs aim at defining the ontology requirements, DROPs are focused on finding the best configuration for being recurrently applied in the domain. Thus, a one-to-one relation (with each CQ attached to a unique DROP and vice-versa) could not be the best organization for all the fragments, and should be reviewed as described in the next step. Moreover, when the core ontology CQs are not available, some reengineering may be needed, for making explicit the ontology requirements before trying to fragment the ontology model into DROPs.

The complexity of the fragments can vary depending on the problem / solution that they are addressing. Sometimes, a fragment that is a candidate for becoming a DROP contains only two related concepts; in other situations, it can contain a complex combination of concepts and relations. Sometimes the same fragment gives rise to two (or more) variant and alternative patterns; and sometimes a pattern is structurally open (Partial DROP) in order to be completed by another DROP or FOP. Complete self-contained fragments are candidates for Complete DROPs; fragments that need to be completed by other DROPs or FOPs are said Partial DROPs.

C. Review the Model Fragments and Select the DROPs

Fragments too big, addressing more than one modeling problems (for example, connected to unrelated CQs), should be analyzed to check if they would be better handled in distinct DROPs. Domain aspects, as well as foundational aspects should be taken into account in this analysis. For example, suppose that the problems being addressed by the fragment are too interrelated, so that the resulting DROPs should always to be used in conjunction. In this case, it is better to maintain the whole fragment in a single DROP. Otherwise, i.e., if each of the two parts can be used independently of each other, it is better to break the fragment into two DROPs.

Fragments too small should also be analyzed to check if they really address relevant problems. If not, then the fragment does not deserve to become a DROP. Moreover, if the problem being addressed is too interrelated to another problem, then consider merging it with the other DROP on which it heavily depends.

A complementary approach is to use FOPs for helping to define the boundaries of DROPs. Since a FOP is applied to solve general modeling problems, many of the domain problems solved by DROPs can be mapped to an underlying FOP structure. Thus, often, a DROP is delimited as the application of a FOP for solving a domain-related problem. This is, however, not a strict rule, since a DROP can be structurally open in a way that it should be completed by another DROP or FOP (thus not directly fitting in any FOP). In other cases, a DROP can apply more than one FOP.

Alternative and useful variants of the same fragment can also be considered. Frequently, a modeling fragment can be represented in different ways, depending on possible variations on what an ontology engineer may want to represent in the target domains. For example, relational properties between two entities can be modeled: by representing a *relator* with the associated *mediation* relations connecting the relata; or by simply representing a *material relation* connecting the relata. In an analogous manner, different domain-level modeling problems can give rise to different alternative DROPs.

D. Pack the DROP with its associated Useful Information.

Information for locating, understanding and using the DROP needs to be attached to it. This includes: name, intent, rationale, CQs addressed by the DROP, the conceptual model fragment, axioms, related patterns (mandatory or optional), and definitions of the types of entities considered in the DROP.

4.4.2 The SEON Core Layer Current Version

We have applied the approach described above for extracting DROPs from the core ontologies and then we have organized them in OPLs. This section presents the SEON Core Layer focusing on the main aspects related to this thesis. The complete diagrams, descriptions, definitions among other information can be found in the SEON Specification[6]. Currently, SEON has only SPO as core ontology.

The Software Process Ontology (SPO) aims at establishing a common conceptualization on software processes. It has 76 concepts and 81 relations organized in 13 sub-ontologies, covering aspects related to Standard, Intended and Performed Processes and their activities, artifacts handled, resources used, procedures adopted, and stakeholders' participation. It imports notions related to organizations, projects and personnel from the Enterprise Ontology (EO) (FALBO et al., 2014).

Figure 4.3 shows how SPO is organized. For aiding identifying concepts and packages, we use the following color scheme to differentiate the main elements in SPO: yellow is related to processes and activities; green refers to artifacts; blue concerns stakeholders; purple is related to resources, such as hardware and software, and pink refers to procedures. Gray is used for external concepts and packages. Darker colors represent types. Later in this chapter, when presenting domain networked ontologies, white is used for domain concepts in domain ontologies.

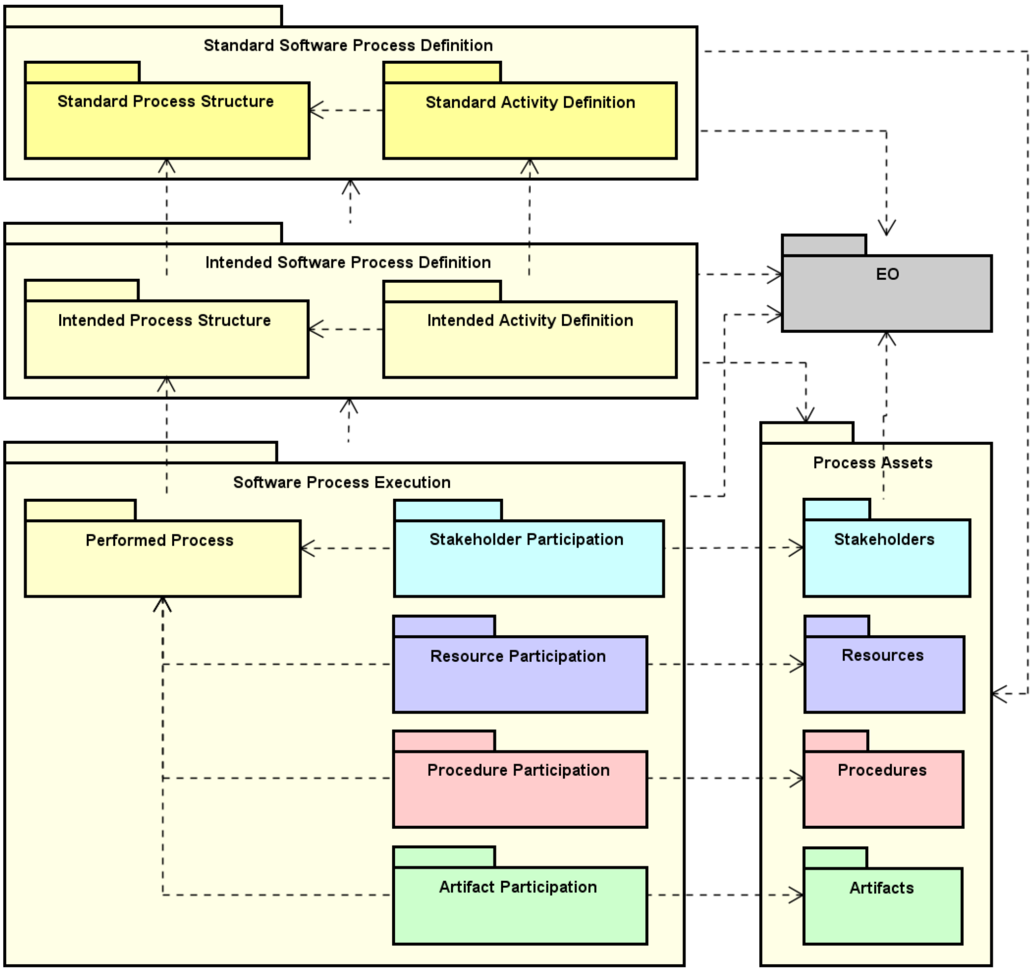


Figure 4.3. SPO Modularization.

[6] SEON Specification is a navigable specification providing a complete reference of the ontology network, as discussed in Section 4.6. It is available at dev.nemo.inf.ufes.br/seon.

SPO is organized in four main packages:

- Standard Software Process Definition package: refers to generic processes (*Action Universals* in UFO) defined by an organization, establishing the basic requirements for intended processes to be performed in that organization. It is further divided in two sub-ontologies: *Standard Process Structure* and *Standard Activity Definition*.
- Intended Software Process Definition package: relates to processes intended to be performed (*Intentions* in UFO) in the context of a specific project or organizational area. It is further divided in two sub-ontologies: *Intended Process Structure* and *Intended Activity Definition.*
- Software Process Execution package: concerns already executed processes (*Actions* in UFO) and the assets' participations (*Participations* in UFO). It is further divided in five sub-ontologies: *Performed Process* (see Figure 4.5), *Stakeholder Participation*, *Artifact Participation*, *Resource Participation* and *Procedure Participation*.
- Process Assets package: refers to the assets handled by a process or it activities, and is decomposed in four sub-ontologies: *Stakeholders*, *Artifacts*, *Resources* and *Procedures*. ***Stakeholders***, ***Artifacts***, ***Hardware Equipment***, ***Software Products***, and ***Procedures*** are considered process assets. These notions, or their types, are related to (***Standard***, ***Intended*** or ***Performed***) ***Activities*** for defining the subject processes in the proper level.

The complete documentation of SPO, including packages, diagrams, descriptions and definitions, is part of the SEON Specification. In this section, we focus on the ontology portions that are germane to our discussion. In the diagrams presented in the sequel, we have preserved the grounding concepts and relations, i.e., the notions in which the concepts / relations are based (concepts extended or relations extended / redefined). In the case of SPO, these grounding concepts are mostly from UFO-B and UFO-C.

Figure 4.4 presents a brief SPO fragment illustrating the main relations between the three process levels: *Standard*, *Intended* and *Performed*. A ***Process Universal*** is a *Complex Action Universal* representing any generic process in an organization (formalized or not). A ***Standard Process*** represents a generic process institutionalized in an Organization (a formalized ***Process Universal***), establishing the basic requirements for intended processes to be performed in an organization or in its projects (for example, a standard Development Process, and its activities, to be performed in the projects of an organization). An ***Intended Process*** is an *Intention* (*Internal Commitment*), usually based on a ***Standard Process***, representing the process defined to be performed in a specific project (a Project Process) or organizational area, considering its particularities. Finally, a ***Performed Process*** is a *Complex Action*, already performed in a specific time interval, by some agent, for reaching a goal in a project or organization. Since UFO define *Action* as instance of *Action Universal*, ***Performed***

Processes are considered instances of ***Process Universals*** (usually those well-established, the ***Standard Processes***).

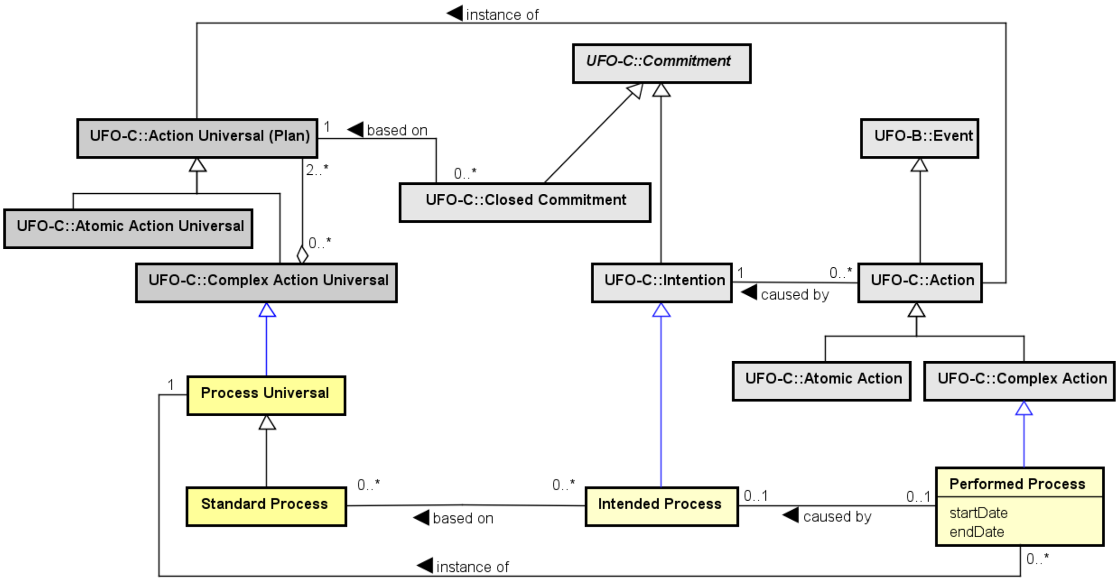


Figure 4.4. SPO: Process Levels Relations.

Although SPO further details each process level (describing the processes types, composition, activities and related assets), the current SEON domain ontologies have mostly focused on the *Performed Process level* for representing SE processes. This is a more natural and intuitive representation, since it describes the processes and activities as they are performed in reality. Concerning the standards harmonization efforts, the *Performed Process level* provides a more adherent representation to standards suggesting or describing which results should be produced and how processes should be performed. Figure 4.5 shows a fragment of the *Performed Process* sub-ontology.

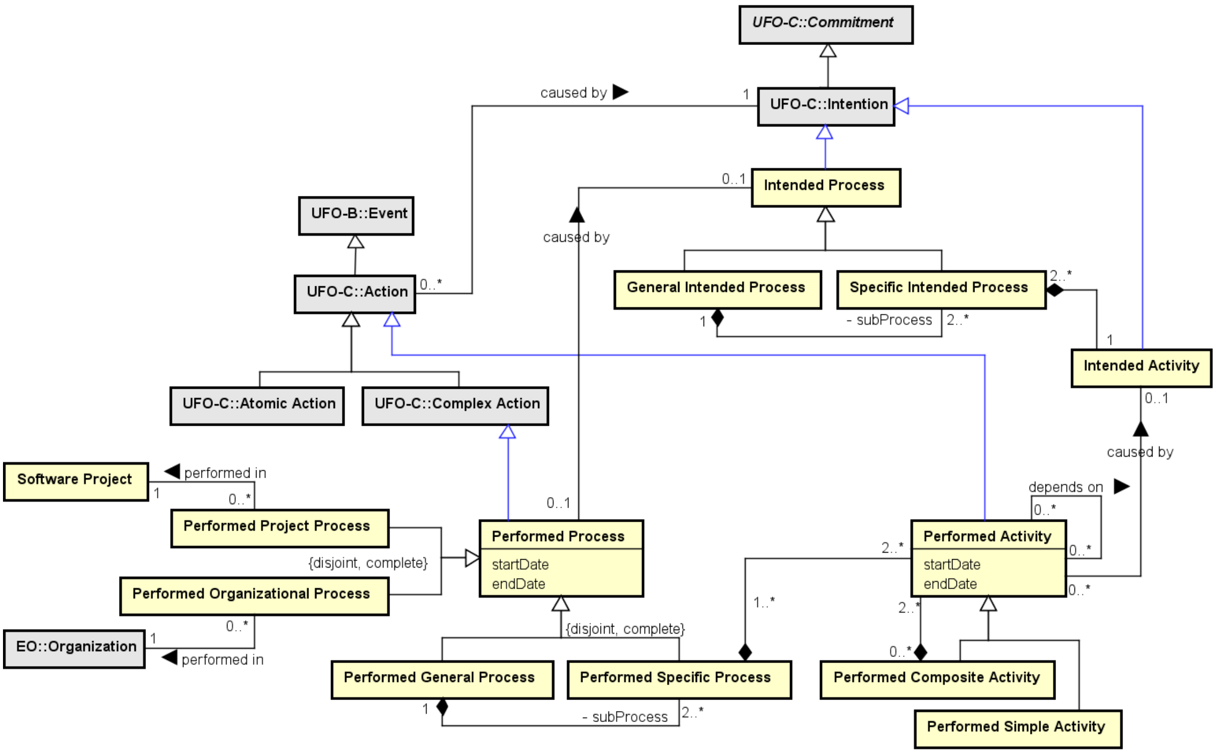


Figure 4.5. A fragment of the *Performed Process* sub-ontology.

Performed Processes and ***Activities*** are actions performed in a specific period of time, in the past. A ***Performed Process*** is caused by an ***Intended Process***, like a ***Performed Activity***, which is caused by an ***Intended Activity***. A ***Performed Process*** can occur in a ***Project***, when it is said a ***Performed Project Process*** (such as a performed Design Process), or in the context of an ***Organization***, when it is said a ***Performed Organizational Process*** (such as a performed Recruitment Process). ***Performed Activities*** are related to diverse process assets for describing the handled artifacts, involved stakeholders and adopted procedures.

Performed Processes can be general or specific. A ***General Performed Process*** (such as a set of related processes performed in an organization) is composed of two or more ***Specific Performed Processes*** (such as an individual process performed in an organization, e.g. *Development Process*, *Project Management Process*) that, in turn, are composed of two or more ***Performed Activities***. A ***Performed Activity***, analogously, can be simple (atomic) or composite (decomposed into smaller activities). ***Performed Activities*** can also depend on other ***Performed Activities*** (e.g. a Review activity that depends on a Documentation activity).

Representation Levels

An important consideration to be made at this point regards the levels adopted for representing SPO concepts, and derived SEON domain ontologies. As discussed, SPO has three process levels: *Standard*, *Intended* and *Performed Process*. The *Standard Process level* contains concepts specialized from UFO universal types (e.g. *Action Universal*), while the *Intended* and *Performed Process levels* specialize from UFO individual types (e.g. *Intention* and *Action*). Since UFO individuals are instances of UFO universals, some of the concepts in these process levels have also a cross-level relation. It is the case of ***Performed Activity*** (an *Action*), which is instance of ***Activity Universal*** (an *Action Universal*).

According to the Multi-Level Theory (MLT) (CARVALHO; ALMEIDA, 2016), since every instance of ***Performed Activity*** (i.e. an activity already performed in reality) is an instance of an instance of ***Activity Universal***, these two concepts can be related by a *isPowertypeOf* relation (see Figure 4.6). ***Standard Activity*** represents the ***Activity Universals*** that are formalized in an organization. Thus, according to MLT, ***Standard Activity*** *categorizes* ***Performed Activity***. It means that the instances of ***Standard Activity*** (the *higher-order type*) can be represented as specializations of ***Performed Activity*** (the *basetype*). This distinction applies to several notions from the *Standard Process level*, the *higher-order types*, and the *Performed Process level*, the *basetypes* (e.g. ***Standard Process*** x ***Performed Process***, ***Artifact Type*** x ***Artifact***, ***Organizational Role*** x ***Stakeholder***, and so on).

Henceforth, since most of the SEON domain ontologies represent performed process types (i.e., types for representing executed processes), while discussing them, we refer to the *Standard Process level* concepts as *higher-order types*, and to the *Performed Process level* concepts as *basetypes*. Thus, in several diagrams representing concepts from core and domain ontologies, concepts from the domain ontologies appear as specializations of the concepts from a core ontology (the *basetypes*). It is just a representation choice, since specializations are more intuitive and easy to reuse than instantiations. Thus, every time a domain concept (e.g. **Quality Assurance Planning**, **Evaluation Report**) is modeled as specialization of a *basetype* core concept (e.g. ***Performed Activity***, ***Document***), it is also an instance of a *higher-order type* (e.g. ***Standard Activity***, ***Document Type***), even if it is not modeled or represented in the same diagram. Figure 4.6 exemplifies the representation levels applied to SEON.

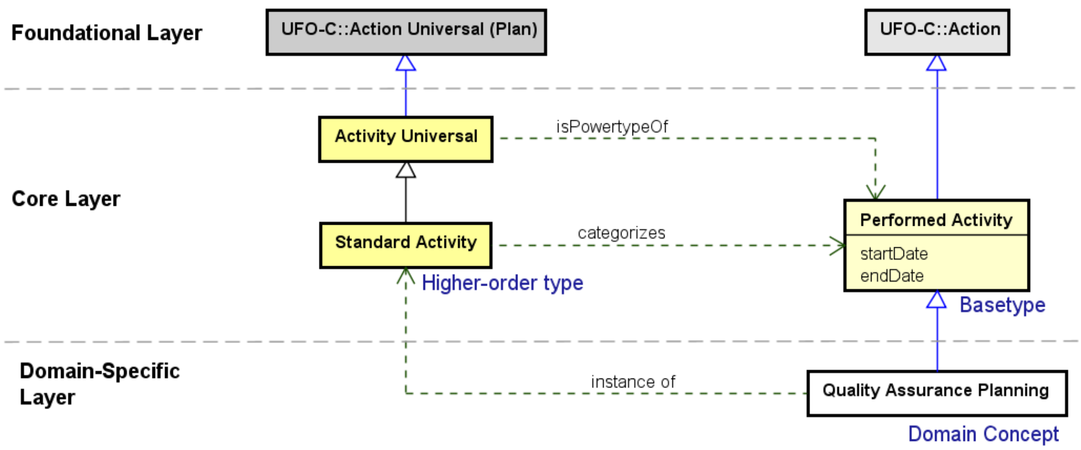


Figure 4.6. SEON: Representation Levels.

4.4.3 SEON Ontology Patterns

The Core Layer models have the purpose of presenting the core ontologies focusing on a better understanding. Thus, they are presented as entire models, showing the grounding concepts (from UFO), such as the SEON diagrams shown in this chapter. However, as we have advocated, organizing them in ontology patterns would improve their reusability and facilitate their application. Thus, the SEON core layer ontologies shall be organized as Ontology Pattern Languages (OPLs). We have partially evolved the original version of SP-OPL (FALBO et al., 2013) to consider the current SPO models and to describe it using ML-OPL (QUIRINO et al., 2017), a modeling language for representing OPLs. We are prioritizing the portions of SPO most used for representing the SEON domain ontologies, especially the ones that can be used for the harmonization efforts. It is a work in progress and this section focuses on the OPL portion containing the DROPs we have been applying for building the current domain networked ontologies.

The OPL Process Models provide the possible sequences of patterns to be applied for creating domain ontologies, guiding the modeling process. Each DROP is represented as an action node (a labeled rounded rectangle) and the flow controls (arrowed lines) determine the admissible sequences. Composite action nodes (labeled rounded rectangles with the hierarchy symbol) denote groups of patterns to be represented in more detailed process models. Figure 4.7 presents the main SP-OPL Process Model. For starting modeling a domain ontology, the ontology engineer should choose between three entry points: EP1, when the domain ontology scope includes standard processes; EP2, when standard processes are out of the scope, but intended processes are to be considered; or EP3, when only performed processes are to be modeled.

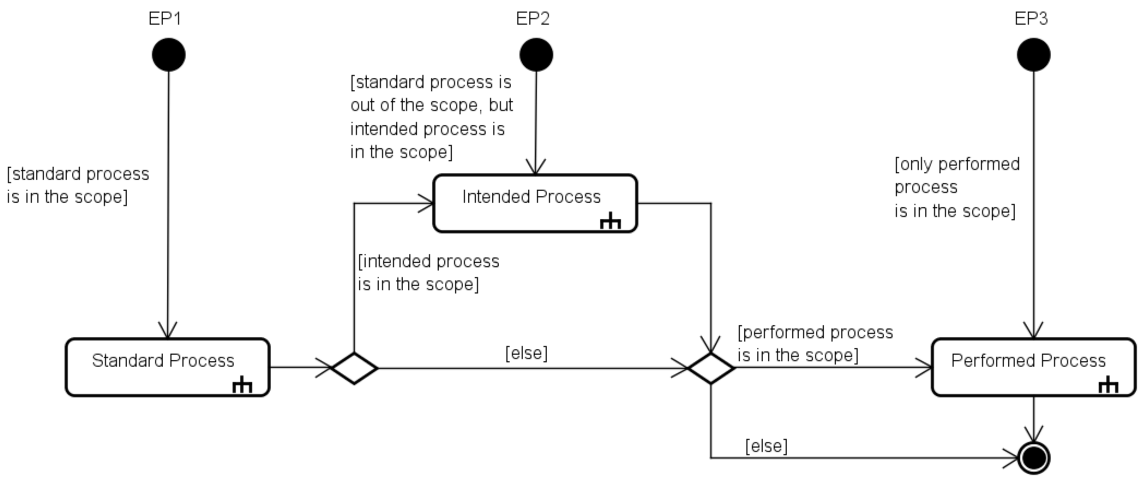


Figure 4.7. SP-OPL: Main Process Model.

In this text, we are focusing on modeling Performed Processes, thus Figure 4.8 presents the Performed Process group of patterns, reached by choosing EP3. Each decision node (represented by a diamond) allows the ontology engineer to choose different paths, for modeling the aspects considered in the scope, in this case, Performed Processes and Activities, and related Artifacts, Stakeholders, Resources and Procedures.

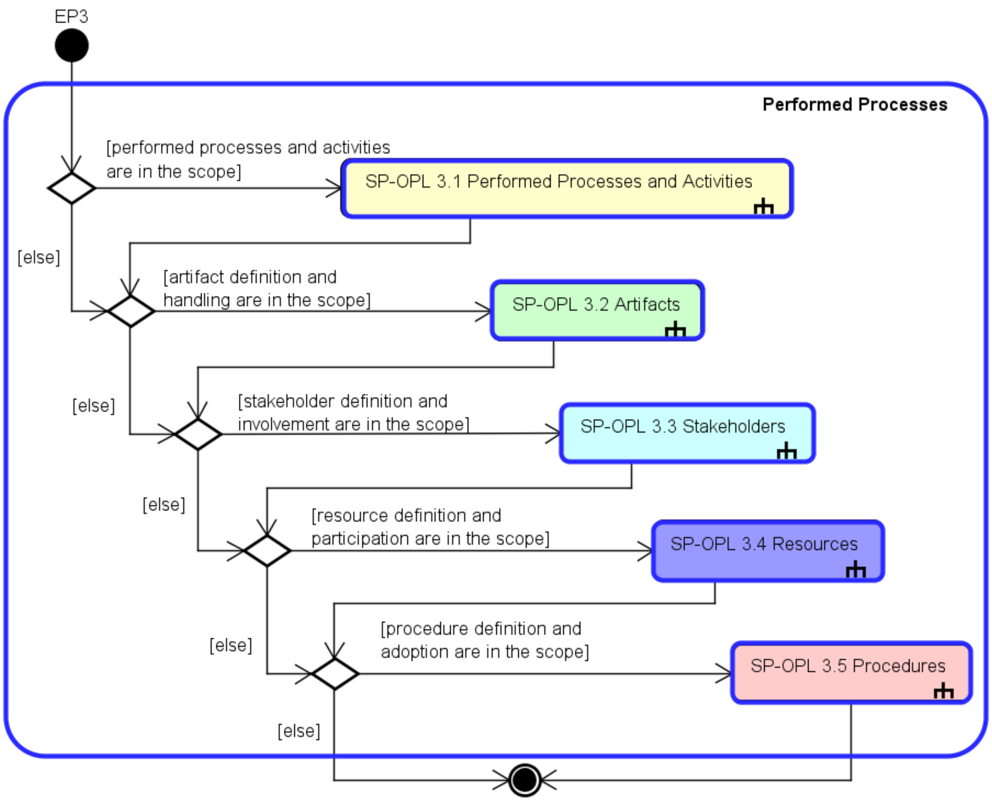


Figure 4.8. SP-OPL: Performed Process Process Model.

By reaching the patterns' group 3.1 (Performed Processes and Activities), the process model in Figure 4.9 is to be used. It presents four DROPs to be applied according to the paths chosen in the decision nodes. The *Process Composition* pattern represents the mereological composition of ***Performed Processes***, defining its specializations ***Performed General Process*** and ***Performed Specific Process***. The *Process-Activity Composition* pattern decomposes the ***Performed Specific Process*** in ***Performed Activities***. The pattern *Activity Composition*, similarly to *Process Composition*, represents the mereological composition of ***Performed Activities***. Finally, the *Activity Dependency* DROP establishes the dependencies between activities.

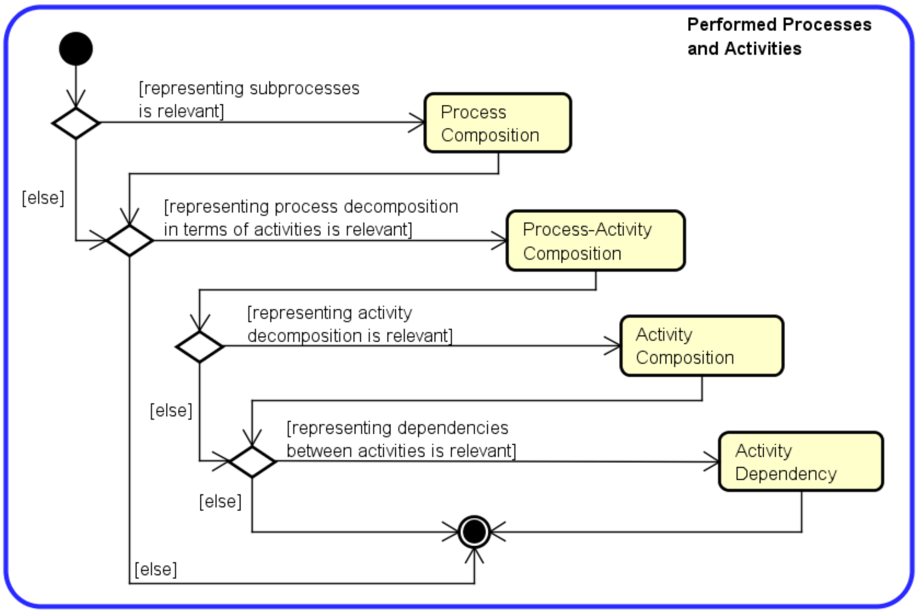


Figure 4.9. SP-OPL: Performed Processes and Activities Process Model.

The other paths of Figure 4.8 can be taken for modeling Artifacts (3.2), Stakeholders (3.3), Resources (3.4) and Procedures (3.5). However, these specific process models are not presented in this text. The main DROPs used for modeling the SEON domain ontologies are modeled in Figure 4.10 and briefly described in Table 4.1. They were extracted from the *Performed Process level* of SPO following the guidelines for deriving ontology patterns from core ontologies. These patterns cover the following aspects: performed processes and activities (3.1), artifacts and how they are handled (3.2), and stakeholder's participation (3.3). Each of the 12 patterns is detached by background labeled boxes.

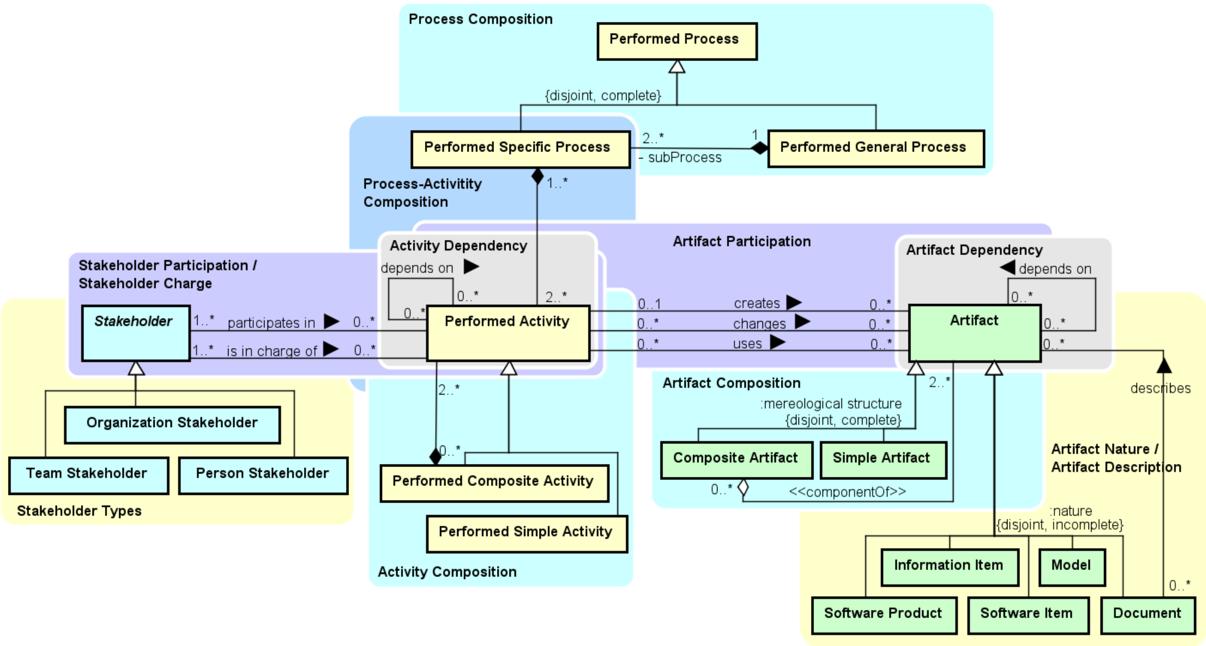


Figure 4.10. SP-OPL Patterns.

Table 4.1. SP-OPL Patterns' Description.

DROP	Intent
Processes and Activities	
Process Composition	Represents the composition of a ***Performed General Process*** in terms of ***Performed Specific Processes***.
Process-Activity Composition	Represents the composition of ***Performed Specific Process*** in terms of ***Performed Activities***.
Activity Composition	Represents the composition of ***Performed Composite Activities*** in terms of other ***Performed Activities*** (simple or composite).
Activity Dependency	Defines the dependencies between ***Performed Activities***.
Artifacts	
Artifact Nature	Describes the different types of ***Artifacts*** according to their nature.
Artifact Description	Defines the description of ***Artifacts*** by ***Documents***.
Artifact Composition	Represents the composition of ***Artifacts*** in terms of other ***Artifacts*** (simple or composite).
Artifact Dependency	Defines the dependencies between ***Artifacts***.
Artifact Participation	Establishes the creation or change or usage of ***Artifacts*** by ***Performed Activities***.
Stakeholders	
Stakeholder Types	Describes the different types of ***Stakeholders*** according to their nature.
Stakeholder Participation	Establishes the participation of a ***Stakeholder*** in a ***Performed Activity***.
Stakeholder Charge	Establishes the responsibility of a ***Stakeholder*** in charge of a ***Performed Activity***.

As the SEON Core Layer evolves and is being specified, it becomes a more robust and reliable basis for the SE general knowledge. The works on the SEMDM ontological analysis, ISP-OPL development and evaluation, and patterns derivation have contributed to this aim. Nowadays, the main source for evolving the core ontologies (and their corresponding OPLs) is the feedback obtained while creating domain ontologies from them. Experimenting new modeling needs and representation solutions in diverse subdomains is contributing to improve the Core Layer, and SEON as a whole. Moreover, as the network evolves, it is able to provide a better support for the development and incorporation of domain ontologies. Facilities such as the ontological and SE grounding and the reusable patterns are subsidizing a faster and qualitative growth of the network.

As discussed, the Core Layer is the heart of SEON and we believe it is the key to leverage SEON's progress in mid/long term. It includes de addition and improvement of ontologies by our group and also by other interested groups.

4.5 Building Domain Networked Ontologies

The domain networked ontologies add domain-specific knowledge to SEON. They must be developed taking advantage of the facilities provided by the network. The grounding in UFO and core ontologies helps classify domain concepts. OPLs guide the way parts of the core ontologies (the patterns) can be reused. Patterns play an essential role by providing extensible modeling fragments with a specification (including competency questions, axioms and other information). Thus, several decisions stem from the network. Of course, these are only instruments to be used / adapted by the ontology engineer to describe the domain knowledge. The contents shall come from reliable sources and the ontology engineer must seek for the better representation of it in reality.

Subsection 4.5.1 presents some guidelines for using the core level patterns in the development of the domain networked ontologies. Section 4.5.2 briefly introduces the domain ontologies currently in the SEON Domain Layer, and, as an example, presents in more details the Quality Assurance Process Ontology. Finally, Section 4.5.3 discusses how new domain networked ontologies can be integrated.

4.5.1 Using Patterns to Build Domain Networked Ontologies

In SEON, the domain networked ontologies are organized according to the subdomains they represent. However, different ontologies can be defined in the same subdomain for describing different aspects (e.g. structure, processes, level of detail). Once the subdomain is chosen and the scope delimited, the steps enumerated below can be performed in a pattern-oriented approach to the domain networked ontology development. It is important to emphasize that we are not proposing an Ontology Engineering method, but only providing some guidelines to be applied while developing ontologies in the network (adapted from (RUY et al., 2017)).

1. **Define the OPL to be used.** Considering that multiple OPLs may be available in the core layer, the ontology engineer shall choose those containing the patterns that provide modeling solutions for the defined scope.
2. **Choose the Entry Point.** For each OPL, the ontology engineer shall define from which entry point to start, according to the OPL process models and the domain ontology scope. She should make sure that the selected EP conducts through the patterns needed for modeling the domain ontology scope.
3. **Select the applicable DROPs.** From the entry point, the ontology engineer shall go through the OPL paths, defining the applicable DROPs considering the domain ontology modeling needs. By selecting a pattern to be applied, many modeling decisions can be expedited, and more elements can be reused (e.g., model fragments, CQs, axioms and concept definitions). For each modeling problem, the ontology engineer shall look for a DROP that can be reused to solve it. A pattern can be selected by looking the pattern’s intent, model fragment and CQs.

4. **Apply the selected DROPs in the domain model.** When a DROP matches the problem at hand, it shall be selected and the corresponding model fragment added directly to the domain model. Typically, the DROP fragment is extended in the domain model, creating new specialized concepts and relations. This reuse approach allows modifying the original pattern, such as adding new properties to the specialized concepts and restricting cardinality constraints in extended relations. A DROP application can reuse concepts already included in the model. Thus, it is important to check if the concept to be represented is already modeled in the actual ontology or in other SEON domain ontology (from where it can be imported or specialized). Latter, the DROP fragments that are not essential for the domain ontology can be omitted from the diagrams, but the specializations shall be preserved in the model to keep the tracking to the core ontologies (useful for integration). At this point, axioms can also be reused (incorporated or specialized), and CQs can be rewritten for standardization.
5. **Solve domain-specific modeling problems.** If no DROP exists for modeling some parts of the domain ontology, a particular solution shall be sought. In these cases, it is highly recommended to anchor each domain concept in a higher level one, from the core or foundational ontologies. Foundational Ontology Patterns (FOPs) can also be useful in these situations.
6. **Check for consistency.** The ontology engineer shall check if the whole model is consistent and if it covers all the planned scope.

4.5.2 SEON Domain Networked Ontologies

Currently, as Figure 4.2 shows, SEON has 12 domain ontologies regarding 9 SE subdomains. These ontologies cover the technical areas of software development (requirements, design, coding and testing) and four supporting areas (quality assurance, configuration management, software measurement, and project management). Most of them describe the target domain focusing on process aspects; however, some ontologies also include details on domain reference concepts and artifacts. Some of these ontologies were created from previous works and some specifically for SEON. Below, we provide a brief description for each SEON domain networked ontologies:

- ***Software Ontology (SwO)***: SwO aims at defining the distinctions around the notion of software product. Software Products are constituted of software artifacts (software items) of different nature, including software systems, programs and code. Moreover, the related specifications are represented. SwO was developed based on the work of Wang and colleagues (2014) and is grounded in the SPO *Artifacts* sub-ontology.
- ***Reference Software Requirements Ontology (RSRO)***: RSRO (DUARTE et al., 2018) aims at being a reference for software requirements notions. It is centered in the notion of requirement as a goal to be achieved, and addresses the distinction between functional and non-functional requirements and how requirements are documented in proper artifacts, among others. It is

mainly based on SwO and UFO. Since RSRO provides the technical concepts for requirements, some of its concepts are often reused in other SEON networked ontologies. Moreover, it is the core of the ***Requirements engineering Ontology subNetwork (ReqON)***, which includes also the other three ontologies for the requirements domain (RDPO, RRO and GORO), presented in the sequel.

- ***Requirements Development Process Ontology (RDPO)***: RDPO aims at representing the activities, artifacts and stakeholders involved in the Requirement Development Process, including the main activities, artifacts and stakeholders for requirements elicitation, conceptual modeling, requirements documentation, verification and validation, and agreement. It is highly integrated to RSRO, since it reuses many requirements-related concepts.
- ***Runtime Requirements Ontology (RRO)***: RRO (DUARTE et al., 2016) focuses on the notions related to the use of requirements at runtime. It was built on SwO, RSRO and UFO, and has two sub-ontologies. The *Program Execution* sub-ontology focuses on capturing the ontological nature of program execution, which is the basis for the use of requirements at runtime. The *Runtime Requirements* sub-ontology addresses the use of runtime requirements artifacts by programs in execution.
- ***Goal-Oriented Requirements Ontology (GORO)***: GORO (NEGRI et al., 2017) was built on RSRO and UFO, and intends to represent the nature of the main concepts and relations surrounding Goal-Oriented Requirements Engineering (GORE) approaches (VAN LAMSWEERDE, 2001).
- ***Design Process Ontology (DPO)***: DPO aims at representing the activities, artifacts and stakeholders involved in the Software Design Process, such as those involved in architectural and detailed design, and design documentation and evaluation. Its notions are in the core of the technical processes in software development, thus DPO reuses some concepts from other SEON networked ontologies, as well provides some concepts to be reused.
- ***Coding Process Ontology (CPO)***: CPO aims at representing the activities, artifacts and stakeholders involved in the Coding Process, for developing a proper code for the software. CPO is in the core of the technical processes in software development, and, like DPO, reuses some concepts from other SEON networked ontologies.
- ***Reference Ontology on Software Testing (ROoST)***: ROoST represents the activities, artifacts, stakeholders, techniques and environments involved in the Software Testing domain, considering only dynamic tests. Again, since testing is a technical process in the software development, ROoST shares concepts with other SEON networked ontologies. ROoST was originally developed by Souza and colleagues (2013) and recently it was integrated into SEON. It is composed of 5 sub-ontologies, namely: *Testing Process*, *Testing Artifacts*, *Testing Stakeholders*, *Testing Techniques* and *Testing Environment*. The main sub-ontology, *Testing*

Process, was developed using SP-OPL. It deals with testing activities for planning, designing and execution of test cases and test result analyses.

- ***Configuration Management Process Ontology (CMPO)***: CMPO aims at representing the activities, artifacts and stakeholders involved in the Software Configuration Management Process. Since CMPO can be applied in the context of several SE subdomains, it describes some general notions applicable for diverse SEON concepts. CMPO is based on the Configuration Management ontology developed by Calhau and Falbo (2012) and was adapted and integrated to SEON. This ontology represents the activities for planning configuration management, controlling changes, establishing baselines, performing audits and product deliveries.
- ***Software Project Management Ontology (SPMO)***: SPMO is defined as a layer over SPO, addressing aspects related to scope, time and duration, and cost estimation of Intended Processes, as well as aspects related to tracking planned versus performed processes. It is divided into three sub-ontologies: *Estimated Process*, *Tracked Process* and *Work Breakdown Structure* (WBS). Some of its key concepts are reused in other networked ontologies (e.g. Project Plan).
- ***Reference Software Measurement Ontology (RSMO)***: RSMO aims at representing the reference knowledge for the software measurement subdomain. It was originally developed by Barcellos et al. (2010) and then integrated to SEON. RSMO is built using both COM (extending the measurement core knowledge for the software realm) and SP-OPL (extending the software process core knowledge for the measurement subdomain). It is organized in five sub-ontologies: *Software Measurable Entities*, *Software Measurement Process*, *Software Measurement Planning*, *Software Measurement Execution*, and *Software Measurement Analysis*.
- ***Quality Assurance Process Ontology (QAPO)***: QAPO aims at representing the main notions involved in a Quality Assurance Process. It describes the main activities for planning and evaluating adherence of products and processes, and controlling noncompliances, as well as the involved stakeholders and artifacts.

For illustrating how to develop a domain networked ontology using the development mechanisms provided by SEON, in the sequel we present QAPO in more details. Figure 4.11 shows the conceptual model of QAPO. The specialized concepts with the *role* stereotype can play different roles as explained in the following.

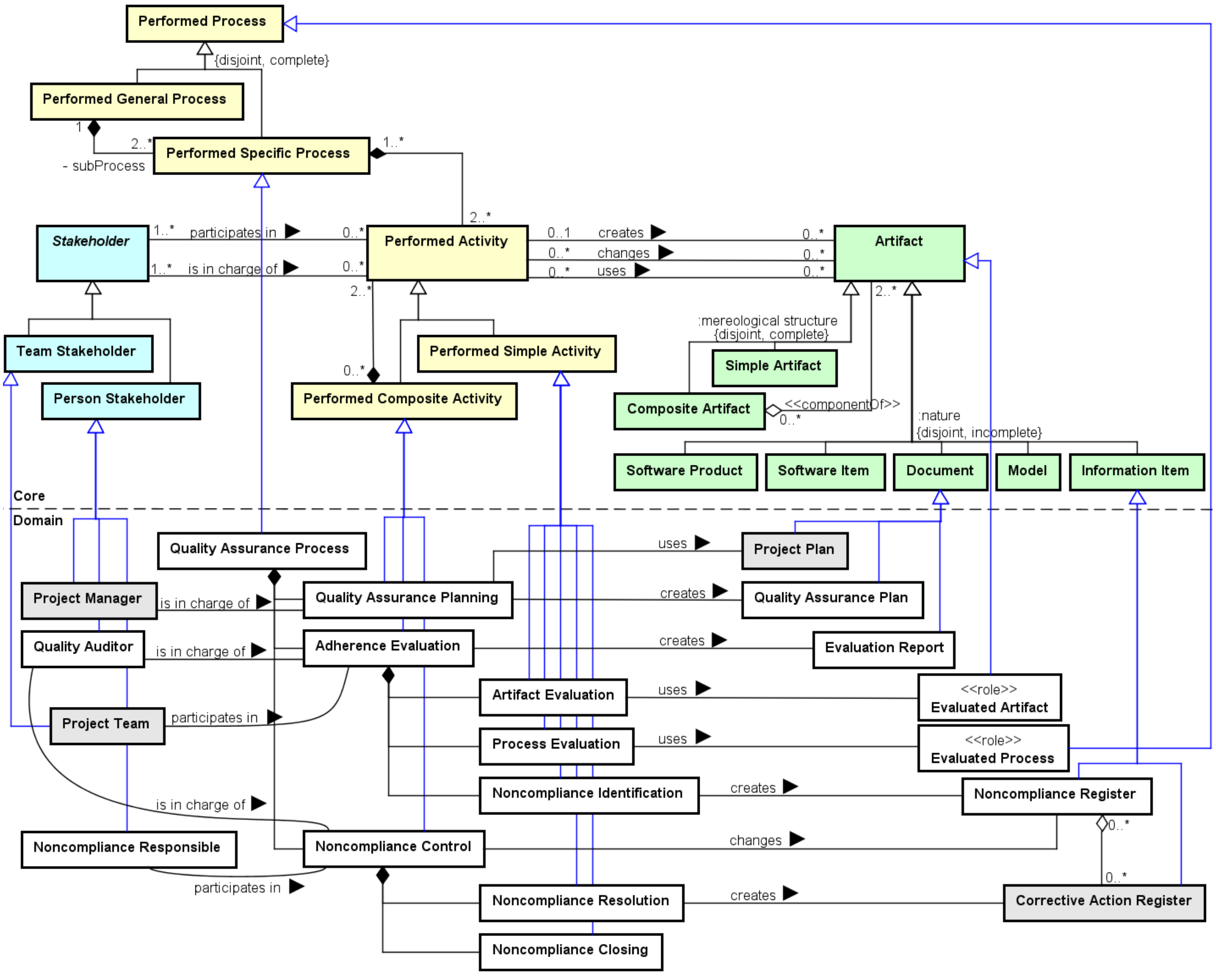


Figure 4.11. QAPO – Quality Assurance Process Ontology.

The **Quality Assurance Process** has the purpose of conducting the activities related to software quality assurance, assessing and assuring adherence of performed processes and produced artifacts to the applicable requirements. It is composed of three main activities:

- **Quality Assurance Planning** plans the quality assurance activities, resulting in a **Quality Assurance Plan**.
- **Adherence Evaluation** objectively evaluates the adherence of processes and products to the applicable requirements, producing an **Evaluation Report** and registering the identified issues. It has three sub-activities: (i) **Artifact Evaluation**, for evaluating the adherence of artifacts; (ii) **Process Evaluation**, for evaluating the adherence of processes and activities; and (iii) **Noncompliance Identification**, for registering noncompliances identified in processes and artifacts.
- **Noncompliance Control** manages the registered noncompliances until their effective resolution. It is decomposed into two sub-activities: (i) **Noncompliance Resolution**, for analyzing a **Noncompliance Register**, and planning and executing the applicable **Corrective**

Actions to its resolution; and (ii) **Noncompliance Closing**, for finishing a noncompliance once it is satisfactorily solved.

The process involves four ***Stakeholders***: **Quality Auditor**, who is responsible for conducting the objective evaluations; **Noncompliance Responsible**, who is assigned for solving noncompliances; **Project Manager**, who is in charge of planning the **Quality Assurance Process**; and **Project Team**, who is involved in the objective evaluations of artifacts and processes related to its project.

QAPO was built from SP-OPL, following the provided guidelines and applying the presented DROPs. Since we are modeling a performed process, we started from entry point EP3 (see Figure 4.7 and Figure 4.8 for details), and then chose the pattern group 3.1 – Performed Processes and Activities (detailed in Figure 4.9). First, we applied the *Process Composition* DROP extending it to create the **Quality Assurance Process** concept as a specialization of ***Performed Specific Process***. Then, the *Process-Activity Composition* and *Activity Composition* patterns were applied for decomposing the **Quality Assurance Process** into the following specialized ***Performed Activities***: **Quality Assurance Planning**, **Adherence Evaluation**, **Artifact Evaluation**, **Process Evaluation**, **Noncompliance Identification**, **Noncompliance Control**, **Noncompliance Resolution**, and **Noncompliance Closing**.

Next, following to the pattern group 3.2 – Artifacts, the types of artifacts involved in the context of a quality assurance process were defined by reusing the *Artifact Nature* pattern, being created three subtypes of ***Documents*** (**Project Plan**, which was imported from SMPO, **Quality Assurance Plan**, and **Evaluation Report**) and two subtypes of ***Information Items*** (**Noncompliance Register** and **Corrective Action Register**, which was also imported from SMPO). The *Artifact Composition* pattern was reused to capture that, in the context of a quality assurance process, a **Corrective Action Register** is part of a **Noncompliance Register**. The *Artifact Participation* pattern was applied for defining specific relations between activities and artifacts. Besides, another specialization of ***Artifact*** was also created: **Evaluated Artifact**, capturing the *role* played by any artifact when evaluated in an **Artifact Evaluation** activity. Analogously, we needed a specialization of ***Performed Process***, **Evaluated Process**, to capture the *role* played by a ***Performed Process*** when evaluated in a **Process Evaluation** activity.

Finally, the pattern group 3.3 – Stakeholders was reached. The pattern *Stakeholder Types* was applied for defining three ***Person Stakeholders*** (**Quality Auditor**, **Noncompliance Responsible** and **Project Manager**), and a ***Team Stakeholder*** (**Project Team**). The last two (**Project Manager** and **Project Team**) were imported from SMPO. The *Stakeholder Participation* and *Stakeholder Charge* patterns were reused for defining the involvement of these stakeholders in the defined activities.

Since Resources and Procedures are not in the scope of QAPO, those pattern groups were not used.

4.5.3 Ontology Integration in SEON

As a network, it is expected that the SEON pieces are integrated. During ontology development, it is possible to observe the integration from a lower to a higher-level, as one of the network premises is the ontological grounding. Thus, a domain ontology should be integrated to (actually, based on) a core ontology, which is integrated to (grounded in) the foundational ontology. However, this vertical integration, although important, is not enough. The ontologies still need to be horizontally integrated, i.e., related ontologies from the same layer shall be integrated, keeping the network consistent.

The SEON integration mechanism adopts some alignment guidelines for matching and integrating domain networked ontologies. When an ontology is to be integrated to the network, firstly, it should be compared with the existing ontologies, looking for equivalent concepts. Since the domain ontologies are produced from (or adapted to) the same basis (UFO and SPO), two concepts can only be considered equivalent if they have the same base type, restricting the search field and speeding up the integration process. For instance, *artifacts* can only be compared with *artifacts*, *performed activities* with *performed activities*, and so on. Keeping two equivalent concepts in the network should be avoided and the most representative one should be reused in all ontologies it is needed. This is the case of the concept **Requirements Document**, defined in RSRO and reused in RDPO, DPO and CPO, as well as **Project Manager**, defined in SMPO and reused in QAPO and other ontologies. If concepts have a partial matching, this could mean that one concept is a specialization or a part of another (e.g., **Configuration Audit Report** from CMPO, that is a specialization of **Evaluation Report** from QAPO). Two concepts from distinct ontologies can also have a relationship between them. In this case, it is worth analyzing if there is a relationship to be extended from the base ontologies or a new relationship should be included in the subdomain ontology. From these matchings, we can determine the correlation level between the ontologies.

Besides the domain ontologies already integrated to the network, we expect SEON to continuously grow by extending the current SE domain coverage and adding new ontologies for other SE subdomains. The SEON integration mechanism has three different ways to incorporate new ontologies into the network, considering the origin of the ontology to be integrated.

Ontologies created for SEON: In a first situation, consider a new ontology that is created for SEON, and thus that is based on UFO and the core ontologies, and also taking other existing SEON's domain networked ontologies into account. Besides the extensions made from the core ontologies, this ontology tends to reuse also the related concepts already defined in the other networked ontologies. This situation occurs in many domain ontologies, and the imported concepts are represented in gray color (for example, two stakeholders and two artifacts in QAPO, as Figure 4.11 shows). This is the best way for increasing SEON, since it reduces modeling and integration efforts, by reusing already defined notions. For assisting this task, the SEON Specification provides a concept search feature, allowing locating concepts by name and definition.

Ontologies created out of SEON, but with the same ground: The second situation occurs when domain ontologies are developed based on SPO and/or UFO, however, independently of other subdomain networked ontologies (usually in a previous or parallel work). In this situation, although the domain ontology to be integrated to the network shares the same basis of the SEON domain ontologies (foundational and core ontologies), some additional integration effort is still required to adapt the common parts focusing on a shared representation. This happened when we integrated ROoST, CMPO and RSMO to SEON. ROoST (SOUZA et al., 2013) was developed based on SPO and UFO, but disregarding the other domain ontologies already integrated to SEON. This way, while integrating ROoST, we had to align it with the other SEON networked ontologies. Figure 4.12 shows a fragment of the integrated model, encompassing elements from four domain networked ontologies: RSRO, DPO, CPO and ROoST. It shows the activities of coding and test case design, and related artifacts. Most of the concepts and relations shown (as the activities for coding and testing) are just imported from their original ontologies. However, some concepts required further decisions. This is the case of the inputs for the **Test Case Design** activity. The **Test Case Design Input** concept is a general role that can be played by different types of ***Artifacts*** that are used as inputs for that activity. In this case, the suitable artifacts are the ones used for creating the code (**Requirements Document** and **Design Document**) and the **Code** itself, giving rise to three new concepts: **Requirements Document as Test Case Design Input**, **Design Document as Test Case Design Input**, and **Code as Test Case Design Input**. These concepts represent roles specializing these three artifacts and playing the **Test Case Design Input** role.

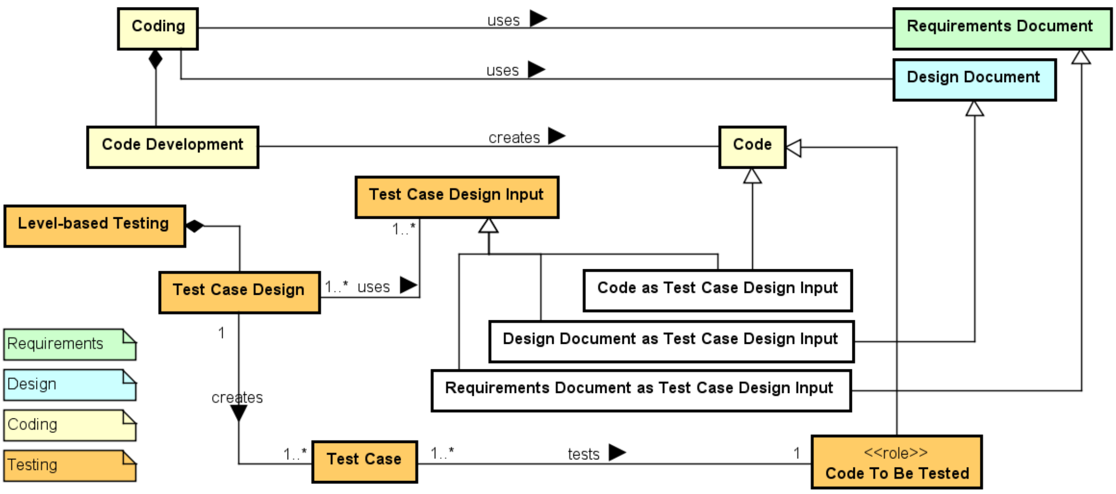


Figure 4.12. An Integrated SEON Fragment.

Ontologies created disregarding the SEON grounding: Finally, the third integration situation happens when external ontologies, developed without taking the core ontologies or UFO as basis, need to be integrated to SEON. In this case, if the ontology can be modified, we should perform an ontological analysis and reengineer it before the integration. By this process, the ontology elements are analyzed and grounded in UFO and core ontologies. The knowledge represented by the ontology is preserved, but the representation is adjusted for a better integration into SEON. On the other hand, if the ontology cannot be modified, we have to make the necessary links and adaptations only in the SEON side. In this situation, techniques for ontology alignment, as discussed in (SUÁREZ-FIGUEROA et al., 2012), apply. Currently, we do not have any external ontology integrated to SEON.

It is worthwhile to say that these are initial guidelines. Although we are already using them (first and second situations), they need to be improved and better specified as the network grows.

4.6 SEON Specification

Building an Ontology Network involves various aspects regarding the creation, integration and evolution of the networked ontologies. Providing an effective access to the network content is essential for its application and improvement. An ON should be presented considering different perspectives and providing useful information. Since we are working with reference ontologies, the ontology diagrams should be accompanied with further information about their concepts, relations and other connections. Moreover, the network documentation shall be kept always accessible and updated.

Documenting an ontology is a laborious work, which increases when considering ontologies in a network. During the network evolution, adding, integrating or improving an ontology potentially affects other ontologies. Manually keeping all this information available in an accessible format is a complex task. Thus, we have developed a transformation tool, able to collect data from our ontology models (currently built using Astah[7]), and transform it into a HTML specification.

SEON Specification is available at dev.nemo.inf.ufes.br/seon/. As Figure 4.13 shows, in left-hand side, there is a menu providing access to the networked ontologies, organized per network layers, and to some features to better explore the network. Each ontology is presented with an introductory text, its related ontologies, its diagram(s) and associated description(s), and concept definitions and other technical details (such as supertype, and relations). The features include: (i) Concept Searcher: a search engine for finding concepts by name and definition; (ii) Network Graph: the network visualization as a graph; (iii) Network Stats: presenting some network statistics; and (iv) Operational Version: a (simplified) operational version of the whole network in OWL.

[7] http://astah.net/

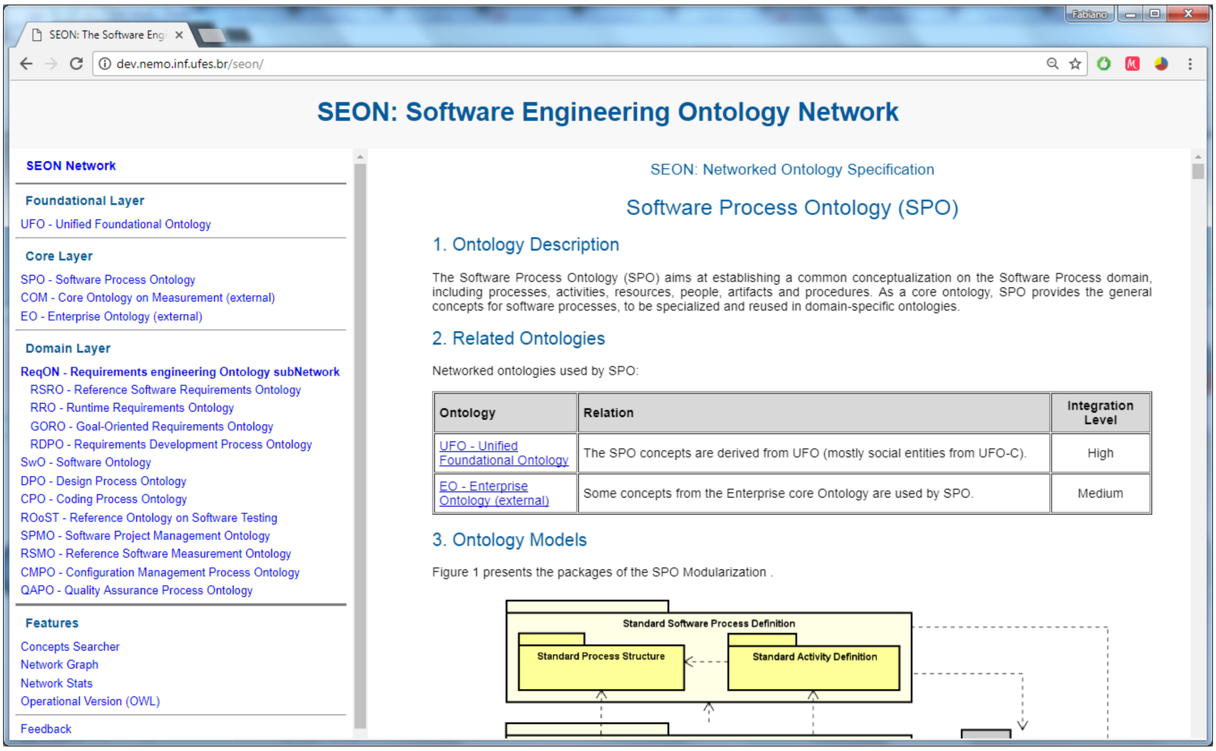


Figure 4.13. The SEON Specification.

The development of such tool helped us to improve SEON in some respects. First, we developed a metamodel for ontology networks to better understand the information handled. This metamodel is to be populated and used to extract data for building the specification and analyzing some information. To automatically extract information from our models, we had to organize them in a uniform way. Thus, we have defined a more precise organization for the network modules (ontologies, sub-ontologies). We also described all modules, diagrams and concepts for providing readable information for the user.

During the transformation process, we have included some consistency verifications for the network to check if all concepts have complete information regarding description (concept definition), ground (specialization from a core or foundational ontology), relations (cardinalities and source and target information) and source (ontology of which the concept is part). These verifications have helped us to fix some problems introduced during ontology integration and to visualize some improvements to the network.

Besides the HTML pages, the tool also produces a preliminary operational version of the network in OWL. It is a lightweight operational ontology, since it is created directly from the reference ontology models without the addition of any axiom. There is a single OWL file for the whole network, where the classes are defined considering their terms, relations and specializations, and some additional information given by the models, such as source ontology and definitions. We expect to improve this feature to use it for performing some reasoning on the network in the future.

Presenting SEON in different perspectives is important for visualizing different characteristics of the network, helping to examine and improve it. For example, the network data is also being used for generating a graph of concepts, as Figure 4.14 shows. The graph presents the concepts grouped according to their relations (very close to a grouping by source ontology). Each ontology is represented in one color (e.g., SPO concepts are in light green). The size of the nodes reflects the number of relations (including specializations) of a concept (e.g., the concepts ***Performed Composite Activity***, ***Performed Simple Activity*** and ***Stakeholder*** are the ones with most relations, thus the biggest ones). Black edges represent binary relations and blue edges, specializations. From this graph, we can better analyze aspects such as integration level, concepts reuse, concepts grouping, and network gaps. It helps planning the next network evolutions.

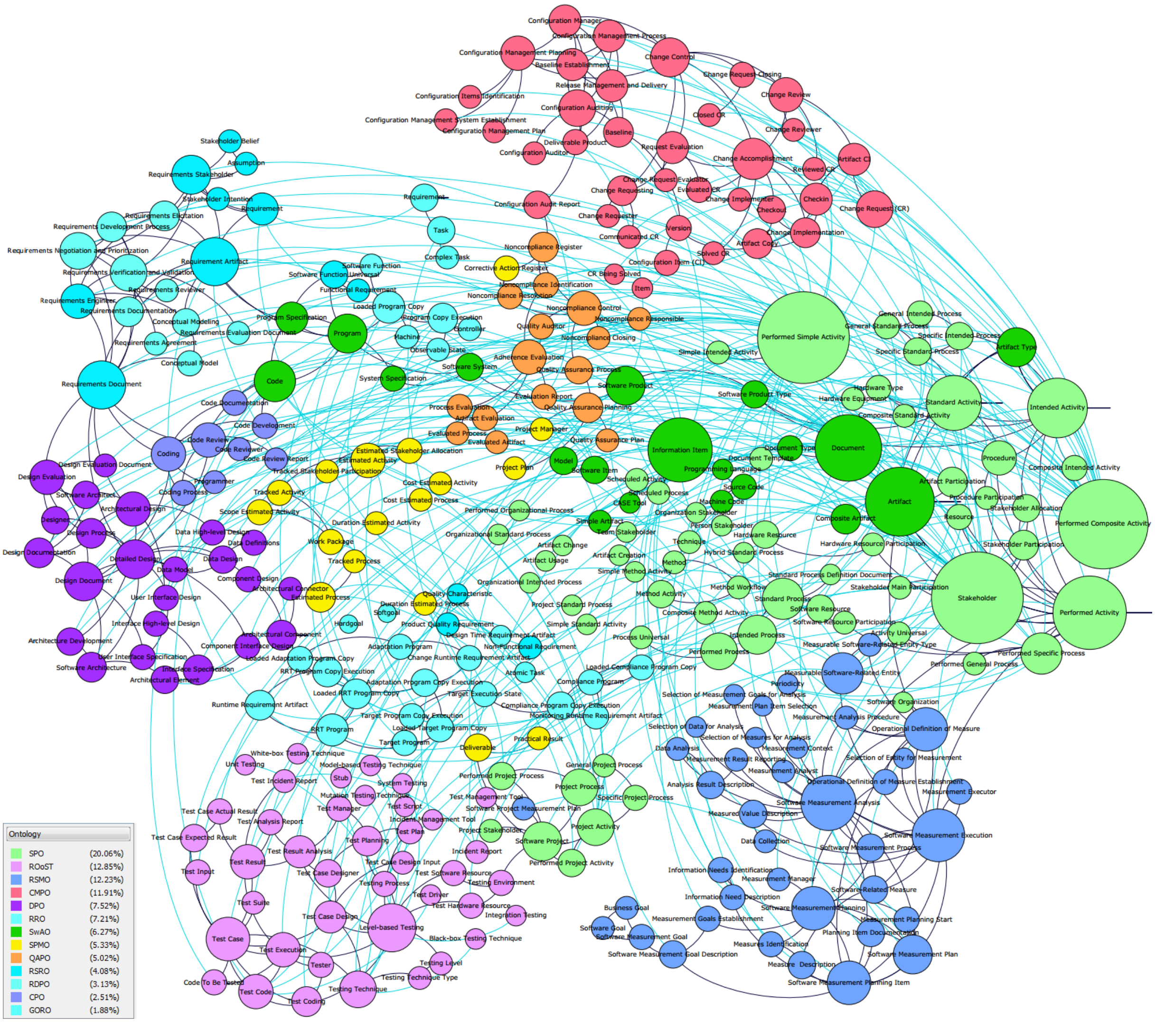


Figure 4.14. SEON Concepts' Graph View[8]

In sum, the specification eases understanding, using and improving SEON, especially for people from other research groups and industry. We have a faster and reliable way to publish SEON new versions.

---

[8] Higher quality graph accessible at dev.nemo.inf.ufes.br/seon/graph

Metadata processing has a great potential for keeping the network consistent and for providing valuable information. More than access, SEON Specification is an important instrument for the network evolution.

4.7 Related Work

This chapter describes many different aspects related to SEON. During the research, we have analyzed some related works covering these aspects. The most important ones are discussed in the following.

Regarding the ontologies aiming at covering a large extension of the SE domain (MENDES; ABRAN, 2005) (SICILIA et al., 2005) (WONGTHONGTHAM et al., 2009), in general, they present many concepts usually based on acknowledged references such as SE books or reference models (e.g. SWEBOK (BOURQUE; FAIRLEY, 2014) and Sommerville's book (SOMMERVILLE, 2004)). Comparing to SEON, the first notable difference is the source for building the ontologies. These Generic SE ontologies are usually based on a few number of sources, in some cases nearing to transcriptions of the referenced source (MENDES; ABRAN, 2005). Contrariwise, each SEON ontology has been built based on a wider set of references, often considering books and standards of the specific (sub)domain. Besides that, knowledge from the base layers is one more source for building the networked ontologies. A second difference regards modularity, since the networked ontologies, even integrated, can be seen, and used, as separated ontologies. Finally, the most important difference regards the mechanisms provided to build SEON incrementally, supported by the foundational and core layers and their patterns. In sum, SEON design considers important characteristics of "beautiful ontologies", as discussed in (D'AQUIN; GANGEMI, 2011), such as: having a good domain coverage; providing relevant reusable distinctions; considering international standards; being modular; being formally rigorous; capturing also non-taxonomic relations; and reusing foundational ontologies.

Taking the support for building integrated ontologies into account, the ISO Ontological Framework (HENDERSON-SELLERS et al., 2014) is intended to be used for harmonizing ISO standards of the SE domain. The goal is to establish a basic set of definitional ontologies, which can be used to derive more specific ontologies, meant to address different SE subdomains. Their structure resembles a tree, and, in some respects, an ontology network. It is characterized by the definitional ontologies and their derived integrated ontologies. The main differences to SEON regard the nature of the ontologies used for grounding the framework and the ontology derivation mechanisms. While the ISO framework has as its basis a SE metamodel as definitional ontology, SEON uses a foundational and core ontologies to provide the necessary (general or SE-specific) ground for the domain ontologies. While the ISO framework provides modest mechanisms for ontology derivation, based on concepts specialization and discarding, SEON uses an OPL for providing patterns and guidance for building the networked ontologies. Actually, we have conducted an ontological analysis over SEMDM, the ISO Framework top ontology, and proposed some improvements for the initiative, as discussed in Section 4.4.

Considering Ontology Networks, Suárez-Figueroa et al. (2012) present three case studies in the fishery and pharmaceutical domains. Three ONs were developed using NeOn method and technologies (SUÁREZ-FIGUEROA et al., 2012). In general, these ONs are composed of operational ontologies (expressed in OWL) plus non-ontological resources (such as *thesauri*). Mappings are an important means to relate the networked ontologies. In two of the studies, the network resources were organized according to the ontologies' types and levels, considering general ontologies (e.g., upper level ontologies or ontologies for time and objects) as independent of the focused domain, and ontologies as references for the domain and basis for providing concepts or relating more specific ontologies. Although the similarities regarding the generality levels, SEON establishes an architecture with well-defined layers, and is based on ontological foundations and patterns, facilitating the building and integration of new domain ontologies. We should highlight that SEON’s architecture is aligned to the one adopted in the ONIONS Project (GANGEMI et al., 1999) and further with the ontological architecture proposed by Obrst (2010).

Finally, a homonymous proposal, SEON (for Software Evolution ONtologies) (WÜRSCH et al., 2012), aims at creating a family of software evolution ontologies for describing the domain of software evolution analysis & mining software repositories. The ontologies’ concepts are organized as a pyramid with four levels: general concepts, domain spanning concepts, domain specific concepts and system specific concepts. Comparing to our proposal, although we share the idea of abstraction for using more general levels as basis for building the more specific levels, they are not proposing an Ontology Network. Moreover, their work focuses on a specific domain with a defined set of applications, while our SEON is open to the SE domain, being applicable for a wider range of situations.

4.8 Final Considerations

SEON is a Software Engineering Ontology Network designed seeking for: (i) establishing a structure for supporting ontologies representing different SE aspects, levels and subdomains; (ii) providing an effective support for developing and integrating new domain ontologies; (iii) being applied for solving semantic interoperability problems in SE; among other requirements and premises.

Accomplishing these objectives is a long-term work. However, we believe the current status of SEON is already a good starting point. The current results, including its architecture, higher-level ontologies, building and integration mechanisms, pattern-orientation, domain ontologies body, and specification, are relevant achievements for promoting its purpose. We have observed that as SEON grows, it is becoming more consistent and able to provide a better support for adding new and improving the current ontologies, in a virtuous cycle. Each new inclusion has pointed out potential improvements in the network structure, mechanisms and contents. In the same way, SEON applications have also contributed to the network evolution, since its use usually provides valuable feedback for improvements.

SEON is the **SE reference knowledge framework** aimed in this thesis to support our standards harmonization approach. Currently, SEON is being used by our approach, Harmony, as support for mapping and integrating standards. However, it can be applied in diverse SE initiatives aiming at harmonizing or integrating models, such as Knowledge Management, Semantic Documentation and Tool Integration (RUY et al., 2016).

Finally, we expect to enlarge SEON coverage by adding new domain ontologies, advance with the defined mechanisms and make it an effective reference for SE knowledge. We also believe it can be applied and increased by other interested groups.

5 An Approach for Harmonizing Software Engineering Standards

This chapter presents Harmony, an Ontology-based Approach for Harmonizing SE Standards. It starts discussing the main requirements used to guide Harmony development. Then it describes Harmony's models, process and guidelines, and its application in a proof-of-concept. Next, Mapper, a supporting tool is introduced. Finally, Harmony is contrasted with related works.

5.1 Introduction

Standards harmonization is a complex problem and requires a semantic oriented solution. Placing multiple standards together and attempting to work harmoniously with them have been a challenge for standards' developers and users. Along the last years, several researches have focused on this topic, proposing diverse approaches for comparing, mapping, and harmonizing SE standards (FERCHICHI et al., 2008) (PARDO et al., 2013) (JENERS et al., 2013) (HENDERSON-SELLERS et al., 2014) (PARDO et al., 2015). These approaches undoubtedly evolved the state of the art of this topic presenting new ways for dealing with the problem. They show that, besides the knowledge and experience in the domain and involved standards, it is essential to follow a systematic and effective approach for harmonizing them. However, as argued in Chapter 3, there is still a research gap regarding the semantic aspects of a desired solution. We believe the next step towards an effective harmonization approach involves the support of a knowledge framework, and a process dealing with the semantics underlying the target standards.

Having these considerations in mind, we defined **Harmony, an Ontology-based Approach for Harmonizing SE Standards**. Harmony is designed based on: our experience with SE standards and conceptual modeling; insights from the proposals we have found in the systematic literature mapping; discussions with experienced research groups; and the incremental development of the approach itself, including its process description, proof-of-concept, a set of initiatives conducted in empirical studies, and the creation of a supporting tool. Some key aspects have guided our proposal, most of them regarding using a domain semantic referential, dealing with models instead of text, addressing multiple harmonization techniques and perspectives, focusing on the meaning instead of mere descriptions, and providing meaningful connections between the involved notions.

Chapter 3 discussed the standards harmonization problem and the proposed approaches in a systematic mapping of the literature. Chapter 4 presented the knowledge framework for providing the domain semantics for such interoperability problem. In this chapter, we deal with the methodological aspects. Section 5.2 describes the essential characteristics the harmonization approach must address.

Section 5.3 provides an overview of Harmony and introduces its main models and connections. Section 5.4 presents the Harmony Process and guidelines while discussing how it addresses semantic aspects in harmonization initiatives; a proof-of-concept is presented for illustrating Harmony application. Section 5.5 introduces Mapper, a tool developed to support Harmony's mapping activities. Section 5.6 discusses the main results achieved and contrasts Harmony with related works. Section 5.7 concludes the chapter.

5.2 Approach Requirements

Harmony is the main process developed in the Design Cycle (HEVNER, 2007) of this thesis. Its central goal is to provide a semantically-oriented solution for the problem of harmonizing SE standards, relying on the use of a SE ontology network for that. Harmony was developed incrementally, from the first drafts to the improvements supported by a proof-of-concept and empirical evaluations. The version presented in this text does not have the ambition to be a complete final solution for standards harmonization, however we believe the approach is already stable enough to be used by practitioners in real world cases leading to meaningful results and advances in standards harmonization.

Along Harmony development, several decisions have been taken to better define the essential characteristics of the approach. These requirements were based on diverse sources. Some of them came from the literature and research gaps (as discussed in Chapter 3), some from our experience and previous works on ontologies and software processes, and some from lessons learned during the approach development and application. This section presents the characteristics we judge essential for guiding the development of such approach, discussing why they were chosen and what we expect by including them. In the following, each of these Harmony requirements is described and discussed.

Ontologies as interlingua. *The approach shall use domain-related ontologies as interlingua for harmonizing the target standards*. The support of ontologies for harmonizing standards brings several advantages, as discussed in Chapter 3. This requirement refers to the use of ontologies as a semantic referential, having the standards' elements linked to the related ontologies' concepts[9]. We see SE standards harmonization as a semantic interoperability problem. Semantic interoperability, in general, refers to the ability to exchange information based on meaning (VELTMAN, 2001). In several contexts, semantic conflicts arise because the things being integrated (systems, data, models and so on) do not share a common conceptualization. Thus, to solve semantic conflicts, an ontology can be used as an interlingua to map elements from different sources. Moreover, by mapping the standards' elements to the ontologies' concepts, we expect to: (i) attach the domain semantics (from the

[9] In this chapter, to avoid confusion, we are distinguishing the terms used for referring to the notions represented in standards' models and ontologies. *Elements* come from standards (models), *Concepts* come from ontologies. *Notion* is used for generalizing both.

ontology) to the standards' elements, (ii) avoid the creation of mappings from each standard to all other involved standards, by using the ontologies as a referential, and (iii) allow reusing information from mappings already established when the same standard is involved in more than one harmonization initiative.

Use of models. *The approach shall be model-oriented, representing and handling the standards' information as a model in opposition to text.* In our initial experiments, we have harmonized three standards (CMMI-DEV, ISO/IEC 12207 and SWEBOK) considering three distinct domains (Software Requirements Development, Software Construction and Software Testing), producing a harmonized process for each domain. The standards' information was analyzed, compared and integrated into a process by handling the standards' textual contents (with support of some tables). The main lesson of this experiment was that it is not feasible to manipulate a high volume of text without a better support. Depending on the domain complexity and the extension of the involved standards' portions, it becomes an exhaustive and error prone task. Hence, we have decided to, before performing any mapping or integration activity, represent each standard as a conceptual model, composed of elements and relations. In this way, the standards' information is represented in a more concise way (easier to understand and compare each element), the relations between the elements become more explicit, inconsistences and omissions introduced by the natural language can be identified, and the elements and relations are in a more appropriate format to be processed by software, among other advantages.

Use of structural and content models. *The approach shall harmonize standards in two complementary levels: structure and content.* In general, each standard establishes a structure for representing its own content knowledge in a similar way. For example, ISO/IEC 12207 has a structure composed of elements such as *process*, *activity* and *task*, used for representing its 43 processes. In a harmonization approach, it is important to harmonize the standards' structures before dealing with their contents. By doing that, we can avoid comparing incompatible elements (i.e., elements with distinct nature, such as comparing an *artifact* with an *activity*). Thus, we have decided to work with two types of models: structural and content models. The structural models represent the standards' structures, while the content models describe the standards' contents organized according to the defined structures. This approach helps understand the standard's format before modeling its contents, and eases the representation of the contents, since the content model specializes the structural model. It is important to highlight that an alternative for representing the structural models is the use of metamodels in a higher order than the content models, so the elements could be instantiated. However, for improving the solution usability, we have opted to represent the structural models using *basetypes* (in opposition to *higher order types*), since typical attributes and relations can be more easily represented and specialized into the content model.

Elements grounding. *All represented standard's element shall be grounded in a traceable ontological basis.* The ontologies used as interlingua come from SEON, which have all domain and core concepts grounded (direct or indirectly) in UFO. To identify the foundations of the elements, and thus to perform comparisons only between compatible elements, we decided also to ground the standards' elements in the UFO basic distinctions, namely *event*, *object*, and *agent*. As discussed in Chapter 2, *events* (or perdurants) are individuals that happen in time, being composed of temporal parts (such as processes, activities and tasks). *Objects* and *agents* are endurants, individuals that are in time. While *agents* (such as stakeholders and teams) perform actions, perceive events and can have intentions, *objects* (such as artifacts and other information results) are unable to perceive events or to have intentions. Thus, all structural elements (and hence their specialized content elements) should refer to these very basic UFO concepts (called henceforth *ontological types*). These distinctions cover the majority of the elements described by the standards' processes (e.g. processes, activities, artifacts and stakeholders, as exemplified). Other basic distinctions can be added when necessary. By having all the concepts and elements grounded in the same ontological basis, we expect to facilitate the comparison efforts and the identification of inconsistences and divergences regarding the nature of the elements.

Mapping technique*. The approach shall map the standards' elements to the ontologies' concepts before integrating them.* Harmonization often involves integration, which comprises knowing which parts of the standards describe the same aspects. Thus, an essential part of Harmony is mapping the related standards' structures and contents to a knowledge referential. The use of conceptual models grounded in the same basis contributes to the mapping activities allowing for more precise matches. Moreover, the matches created for connecting the elements of a standard to the concepts of an ontology constitute meaningful data for standards analysis, integration and improvement.

Types of matches*. The approach shall provide distinct types of matches for distinct types of connections.* The connection between two notions represented in distinct models cannot be reduced only to *match* or *do not match*. Diverse types of relations can appear during these comparisons, involving equivalent, partial or extension matches, among others. Moreover, notions with different grounds can reveal distinct types of matches. *Events* such as process and activities could be partially matched, while *objects* such as artifacts could be specialized, generalized or even play a role. We believe that providing distinct types of matches favors representing the mappings in a more precise and meaningful way.

Integrated model as result*. The approach main result shall be an integrated model.* There are many ways to represent the results of a harmonization initiative. It could be a standard-like text integrating the involved standards (as Pardo et al. (2013) do) or a table presenting the mappings established (as Salviano and Figueiredo (2008) do). However, we have decided to represent, as the main harmonization result, a conceptual model providing an integrated view of the involved standards (as Jeners et al. (2013) do). In our approach, this model should be defined from the ontologies used as *interlingua*, extended with any necessary standard's element, and linked to the corresponding matches. It is a representative way for accumulating all the handled information. From this model, it is possible to create other representations (such as a harmonized process description or a mapping table), since all the necessary information is preserved. Moreover, this integrated model presents the information in an organized manner, being easier to understand what it is represented.

Once the aforementioned requirements are presented, it is important to highlight that they do not constitute all the requirements for the approach. However, they are the core ones, taken as premises for developing and improving Harmony with the characteristics we have considered essential.

5.3 Harmony Overview

Harmony is our approach for conducting harmonization initiatives through a process guiding the user for representing the involved standards and applying the proper harmonization techniques. Harmony works allied to SEON, which provides the SE reference knowledge as an ontology network. Additionally, the Mapper tool supports Harmony's mapping activities.

Harmony was designed for harmonizing quality standards from the SE area. More specifically, we have tested it only for standards focusing on *software processes* (such as CMMI-DEV, ISO/IEC 12207, ISO/IEC 29110 among others). Although we believe the approach could deal, with a few adjustments, with other types of standards (such as for *software products* or even *system processes*), we have not evaluated it in these contexts. Harmony target audience comprises people with intermediary or high experience with SE standards and conceptual modeling. Even though we have experienced it with students with lower experience, this application required a closer monitoring and support.

Harmony can be used in several situations, including: by standards' developers aiming at harmonizing published or proposed standards; by researchers working on standards' similarities and differences, their semantics, and other aspects; or by standards' users, such as organizations attempting to deploy an integrated solution in a multimodel environment.

The process can be performed in a complete or a partial way, according to the defined purpose. I.e., it allows achieving only structural harmonization or to proceed until content harmonization producing an Integrated Content Model (ICM). Moreover, it can be followed by other complementary activities such as for directly mapping a pair of standards or producing a harmonized process description.

Harmony is built based on three main actions: Modeling, Mapping and Integration. **Modeling** refers to representing the standards as models (in opposition to text) for better dealing with them. **Mapping** regards mapping the standards' models to ontologies serving as a semantic referential. **Integration** concerns building a unified view of the standards by extending the domain view with the necessary standards' specific elements.

Figure 5.1 shows a general view of Harmony, illustrating some central ideas. First, the standards' portions to be harmonized are selected and modeled, representing the standards' information in terms of conceptual models. Then, each element of the standard's model is mapped to a proper concept of a selected view of SEON. Finally, according to the harmonization scope, this view is extended based on the models and mappings, resulting in an integrated model. This sequence occurs in two rounds: first, dealing with the standards' structure and, then, with the standards' contents.

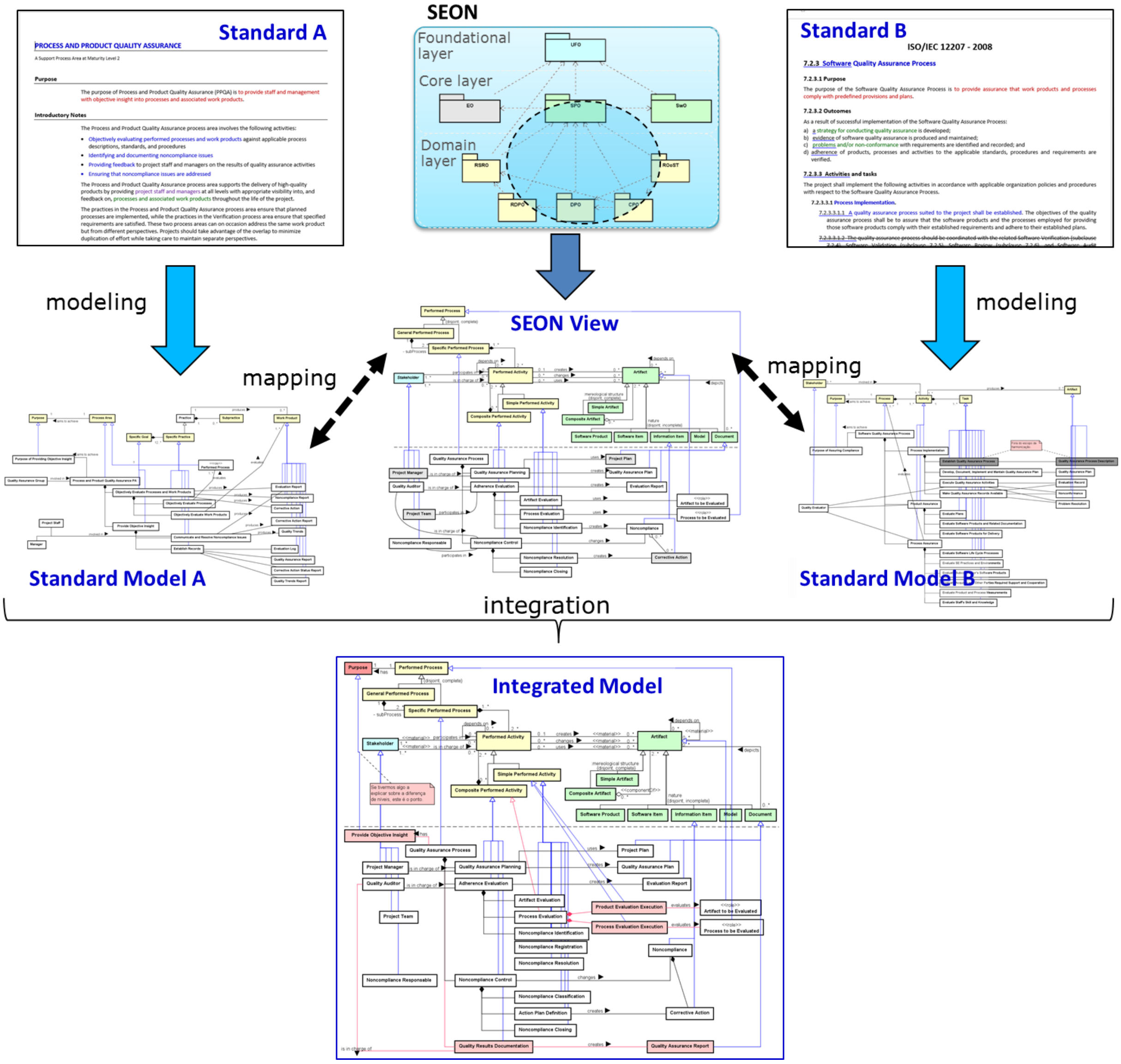


Figure 5.1. Harmony General Scheme.

5.3.1 Harmony Models

An essential characteristic of Harmony is dealing with models for accomplishing its process. Conceptual models are used for representing the standards and the integrated results; ontologies are used as an interlingua, being the basis for the mappings. In this process of handling standards and ontologies for establishing the proper connections and integrated results, some terms are defined for better dealing with the complexity of the activity. It is important to clarify these terms to provide a better understanding of the Harmony process described ahead. Figure 5.2 shows a general view of Harmony models, and their connections.

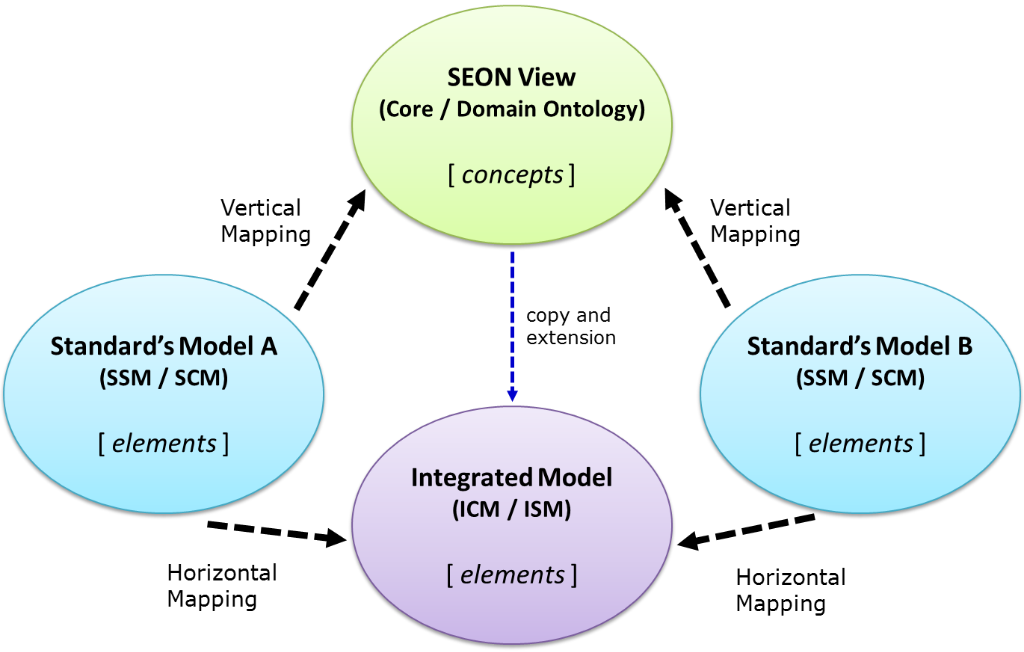


Figure 5.2. Harmony Models.

The standards involved in a harmonization initiative are represented in two types of models: **Standard's Structural Model (SSM)**, which represents the structure of a standard, and **Standard's Content Model (SCM)**, which represents the contents of a standard. Both standards' models are composed of ***elements***. Each element represents a concise portion of information extracted from a standard and is shown as a class in our diagrams. Since the content elements (from a SCM) are specialized from the structural elements (from a SSM), in this relation, the latter are called ***base elements***. For instance, an initiative involving the CMMI-DEV standard would produce a CMMI SSM (with elements such as *Process Area* and *Specific Practice*) and a CMMI SCM (with elements such as *Project Planning* and *Estimate the Scope of the Project*, specialized from their respective base elements in the SSM). Although the figure represents models for only two standards (A and B), the approach supports harmonizing more than two standards.

SEON View is a selection of the SEON portions to be applied in an initiative. This model is composed of a selected set of ***concepts*** and relations from core and domain ontologies. When needed, the terms ***core concept*** and ***domain concept*** are used for differentiating the concepts from the core and domain parts of the SEON View, respectively. Concepts from Foundational Ontologies are not explicitly represented in the approach diagrams; however, the most basic ones are used for grounding the models' elements and the ontologies' concepts, being referred as ***ontological types***. For illustrating, *Requirements Development Process* is a domain concept, specialized from the *Performed Process* core concept. Both concepts, as well as both CMMI elements in the previous example, are grounded in the ontological type *event*, i.e., all of them have an *event* nature, and thus, can be compared with each other. During the harmonization activities, the core concepts of the SEON View are used for structural harmonization (involving the SSMs), while the domain concepts are used for content harmonization (involving the SCMs).

Another type of model produced by the approach is the integrated model. Each initiative can create up to two integrated models: **Integrated Structural Model (ISM)**, a unified view of the structure of the set of standards being integrated; and **Integrated Content Model (ICM)**, a unified view of the content of the set of standards being integrated. They are created as a copy of the corresponding parts (core or domain) of the SEON view and extended by including / modifying / removing elements. As models, ISM and ICM also contain ***elements***. Moreover, the term ***notion*** refers to a generalization of the terms *concept* and *element*.

In Harmony, mappings should be established between models and ontologies (SEON view). **Vertical Mappings** are established having a model as source and an ontology as target, respecting the generality levels. Thus, the mapping between a SSM and a core ontology is a Structural Vertical Mapping; and the mapping between a SCM and a domain ontology is a Content Vertical Mapping. **Horizontal Mappings** are those established between a standard's model (as source) and an integrated model (as target).

Mappings are a set of **matches**. **Simple matches** are binary relations from a source element (always from a model) to a target notion (an ontology concept in vertical mappings or a standard's element in horizontal mappings). While creating a **match**, besides the pair of notions, it shall also be defined a *match type*, a *coverage* and *comments*. **Composite matches** allow aggregating a set of simple matches with the same source, aiming at complementing the coverage of this source element. Figure 5.3 shows a fragment of the Mapper's model, illustrating how we are dealing with mappings and its matches.

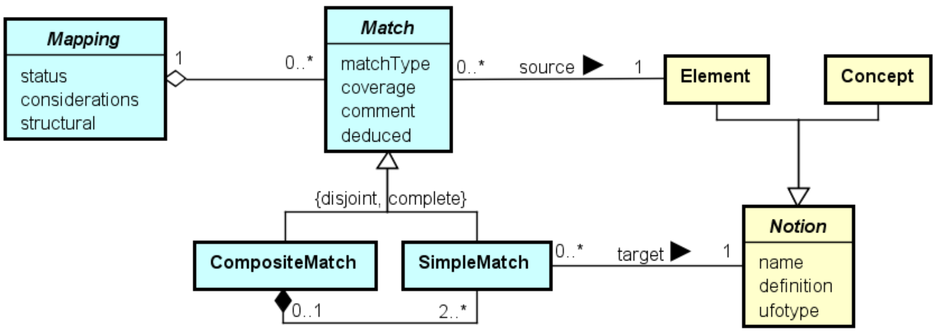


Figure 5.3. Mapping and Matches Representation.

5.4 Harmony Process

Once Harmony main ideas and models are presented, the methodological aspects take place. Figure 5.4 shows an overview of the Harmony process with its main results. Three phases are defined:

1. **Purpose and Scope Definition**, for establishing the harmonization purposes, scope, and the portions of the ontologies and standards to be used;
2. **Structural Harmonization**, for producing the Standards' Structural Models (SSMs), mapping them to the core ontologies and creating the Integrated Structural Model (ISM); and
3. **Content Harmonization**, similar to phase 2, but for dealing with the standards' contents: produces the Standards' Content Models (SCMs), map them to the domain ontologies, and creates the Integrated Content Model (ICM).

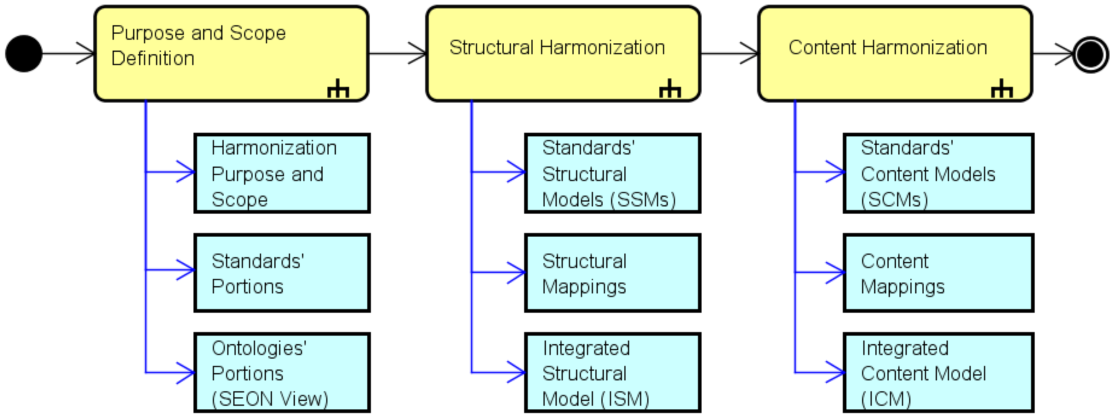


Figure 5.4. Harmony Process.

Phases 2 and 3 are similar regarding their goals, but focus on different aspects: standards' structure and content. Moreover, phase 2 (Structural Harmonization) is a preparation for phase 3 (Content Harmonization). It is important to notice that, although the activities are shown in a sequential order, they can be performed with some iteration, admitting return to activities (and phases) as the process progresses. The results produced in each activity (descriptions, models, mappings, etc.) should be grouped in a harmonization report, organized in sections according to the activities. Examples of harmonization reports are available in Appendix A and Annex A.

The first Harmony complete application was for harmonizing the Quality Assurance processes of CMMI-DEV and ISO/IEC 12207. We accomplished this initiative as a proof-of-concept, which has been updated as Harmony evolves. The final version of the Quality Assurance (QA) Initiative report is in Appendix A. During the forward presentation of Harmony's activities, for illustrating the approach application, we present boxes with some fragments of this initiative.

5.4.1 *Purpose and Scope Definition*

This phase defines the purpose and the scope of a harmonization initiative. As Figure 5.5 shows, it encompasses the definition of the harmonization purpose and scope, and the selection of standards' and ontologies' portions to be used. It is a preparation for the next phases, defining important parameters for the modeling, mapping and integration activities and the focus given to them. Although it is the first phase, it can be revisited for adjustments and detailing.

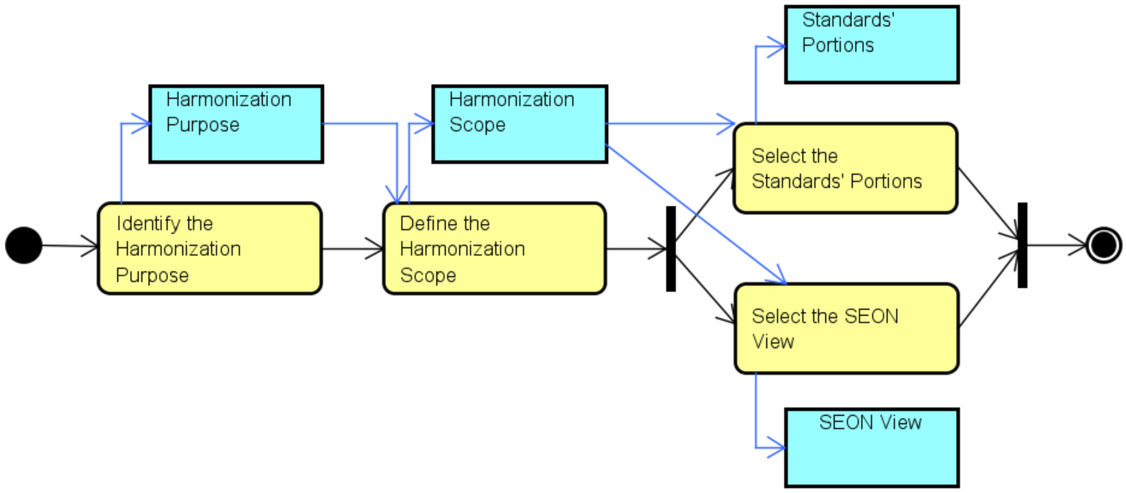


Figure 5.5. Purpose and Scope Definition.

A1.1. Identify the Harmonization Purpose

The harmonization purpose shall be established, making explicit the reasons why the initiative is being conducted. Purpose is crucial for a harmonization initiative, since it defines its intentions and expected results. It can lead to adaptations in the process, establishing how and whether some activities should be performed. For example, if the purpose is to harmonize only the structure of the involved standards, only the Structural Harmonization phase shall be performed; if the purpose involves producing a direct mapping between a pair of standards, or a harmonized process description, specific activities can be added to the process. The purpose must be kept in mind during the harmonization initiative. Typical harmonization purposes include: *deriving an integrated version of the selected standards' portions*, *identifying standards' inconsistencies*, and *establishing a common vocabulary for some domain*.

QA Initiative: Harmonization Purpose
To provide a harmonized model for the Quality Assurance Process of CMMI-DEV and ISO/IEC 12207.

A1.2. Define the Harmonization Scope

The harmonization scope establishes the domain boundaries of a harmonization initiative. It shall be defined selecting the domains (e.g. *Requirements*, *Project Management*) and the specific coverage to be considered (e.g. *only Requirements Development, not considering Requirements Management*; *only*

Project Planning, not considering Project Monitoring and Control). This coverage can embrace even the sum of portions from different domains (e.g. *Requirements Development, from the Requirements domain, plus the Design of Test Cases for Testing Requirements, from the Testing domain*). The scope description should include the parts of the domain(s), and the types of elements to be considered (e.g. *processes*, *activities*, *artifacts*, *stakeholders*, but not *tools*).

Defining the scope is important to delimit the extension of the standards and the ontologies to be considered. The scope will guide on deciding whether a certain information should be included or not in the standards' models for the harmonization initiative. Thus, the defined scope is the basis for the next two activities, where the standards' and ontologies' portions are selected. As the process advances, this activity (and the following two) shall be reviewed, making adjustments for defining the scope more precisely (as well as the standards' and ontologies' portions to be used).

QA Initiative: Initiative Scope
Software Process and Product Quality Assurance, from Planning to Evaluation and Noncompliance Control, including processes, their activities, related artifacts and involved stakeholders.

A1.3. Select the Standards' Portions

This activity states which parts of the standards will be handled. The portions of each standard to be considered in the harmonization initiative shall be defined based on the harmonization purpose and scope. This selection needs to be made precisely and carefully, since each standard presents its own structure and sometimes the target portions are spread along many different sections / processes (e.g., the Test domain does not have a specific process area in CMMI, but information about it can be found in process areas such as *Technical Solution* and *Product Integration*). Moreover, relevant information can be found besides the main standards' processes or sections, such as in standards' glossaries, in generic practices (such as in CMMI) or more specific or general processes (such as in ISO/IEC 12207).

The description of the standards' portions that will be the target of the initiative shall make explicit the processes / sections / practices to be considered. It is important to notice that selecting the standards' portions is not just choosing which sections of each standard will be addressed by the initiative. It is a knowledge-based activity, requiring a preliminary analysis and understanding, and a precise definition of which processes, activities, practices and so on will be considered. It must be highly aligned to the initiative purpose and scope, and accurately described by the user.

For aiding this activity, if there is a SSM available from a previous initiative, it can be useful to understand the standards' structure before selecting the standards' target portions. Moreover, a good practice is to create a document for each standard copying all the selected portions there. It organizes the portions, eases checking, and can be used as basis for building the standards' models.

QA Initiative: Standards' Portions
CMMI-DEV (v.1.3): PPQA Process Area, and related generic practices applied to PPQA, namely: GP 2.2 – Plan the (Quality Assurance) Process, GP 2.4 - Assign Responsibility and GP 2.7 – Identify and Involve the Relevant Stakeholders.
ISO/IEC 12207 (2008): the whole Software Quality Assurance Process (7.2.3), and specific activities from other two processes for dealing with nonconformities control, namely: Quality management corrective action (6.2.5.3.2) from Quality Management Process (6.2.5), and Problem resolution (7.2.8.3.2) from Software Process Resolution Process (7.2.8).

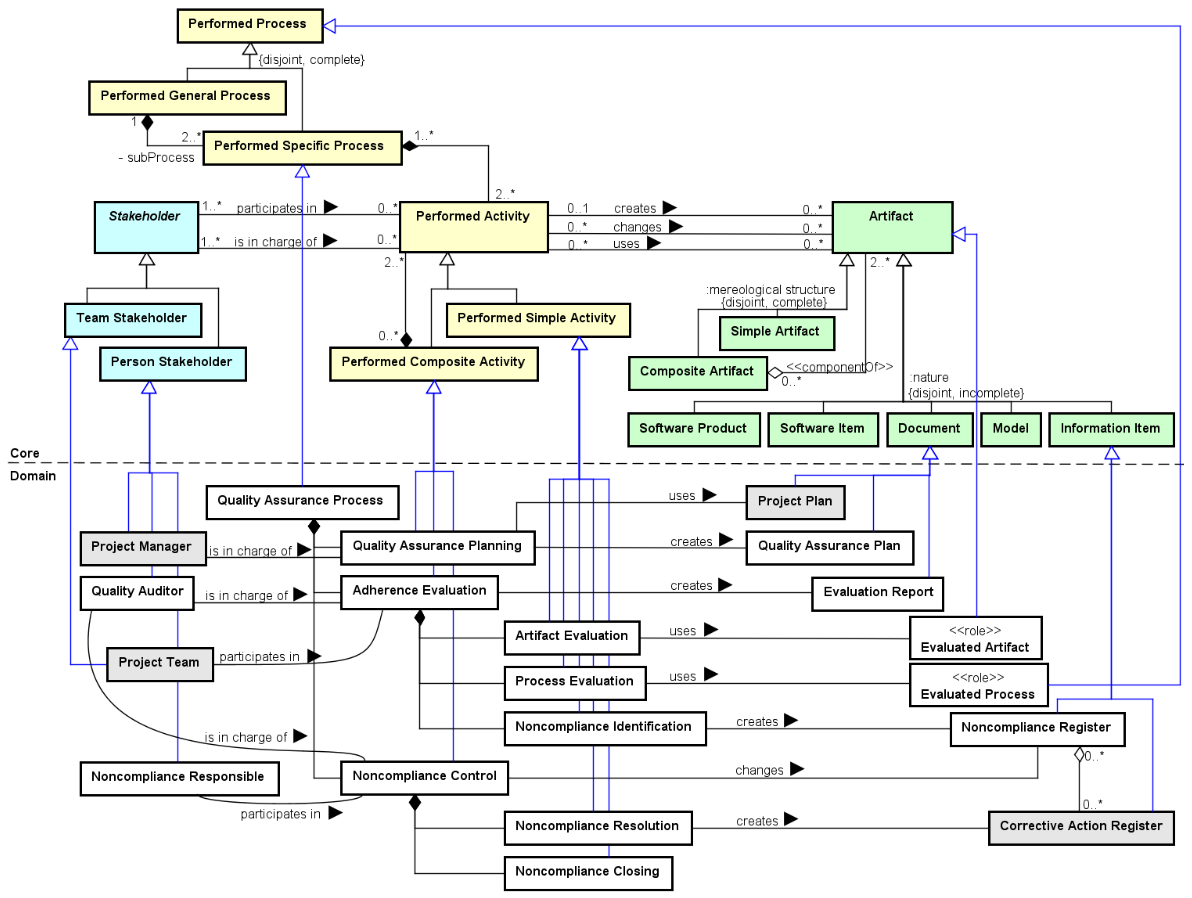
A1.4. Select the SEON View

The SEON View is the portion of the SE ontology network that will be devoted to cover (i.e. to be matched with the maximum of) the standards' elements. It represents the domain knowledge, a semantic reference independent of any standard. The portions of the SEON ontologies to be considered for the harmonization initiative shall be identified, focusing on the ones that can provide useful concepts for the defined scope. Like the standards' portions, this selection should be precise and guided by the initiative purpose and scope. The selection of domain ontologies is related to the domain(s) being considered (e.g., *Requirements*, *Design*, *Testing*), while the selection of core ontologies is related to the types of concepts (structure) to be represented in the resulting model (e.g., *processes*, *activities*, *artifacts* and *resources*). Both are important, since they provide the referential concepts for the harmonization initiative.

Moreover, some desired SEON's portions can be dispersed along different ontologies, not providing a clear complete view of the defined scope. Thus, the portions of each ontology to be considered shall be grouped and reproduced in a single conceptual model, one diagram representing the SEON View to be used. The establishment of the SEON View as a single conceptual model (including core and domain concepts from different ontologies and all proper relations between them) is important to facilitate the mappings and to extend the view to the forthcoming Integrated Models.

QA Initiative: SEON View

SPO (core ontology) portion dealing with processes, activities, stakeholders and artifacts, and the whole QAPO (domain ontology), as shown in the diagram below.



5.4.2 *Structural Harmonization*

This phase focuses on harmonizing the structure of the involved standards. The idea is that, by harmonizing the structures first, the contents will be more easily understood and harmonized. The main results are the Standards' Structural Models (SSMs), their mappings to the core ontologies, and the Integrated Structural Model (ISM).

Figure 5.6 shows the activities comprising this phase. The first two activities are performed individually for each standard, while the two other deal with all the standards together. Since some results of this phase may already be available from previous executions of the approach (such as structural models and their mappings to SEON), some of its activities may be skipped for certain standards. For example, in an initiative harmonizing the standards StdA and StdB, if StdA's SSM is already available and mapped to SEON (1st and 2nd activities), it is only necessary to develop the StdB's SSM and map it to the core ontology. Then, both models should be considered while building the ISM (3rd and 4th activities). Other different possibilities can arise depending on the availability of previous results for the involved standards. Reusing previous results should be considered whenever possible.

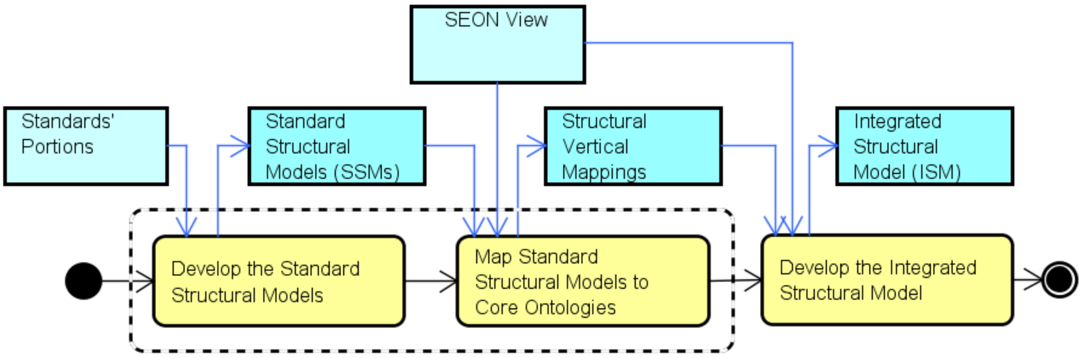


Figure 5.6. Structural Harmonization.

The Structural Harmonization phase works as a preparation for the Content Harmonization phase, since it establishes the base models to be extended and the mapping possibilities. Moreover, at this point, besides identifying the standards' structural similarities, it is possible to observe also some divergences, as discussed in the next activities.

A2.1. Develop the Standards' Structural Models

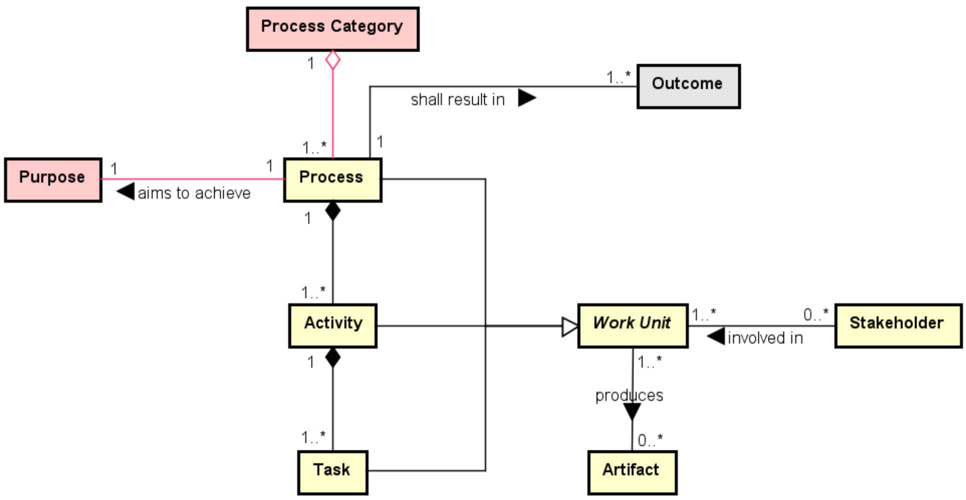
Each standard has an implicit model. This activity aims at making the structural models of the standards explicit by creating the Standards' Structural Models (SSMs). For each standard, a SSM shall be developed (or reused from a previous initiative). It is a generic model extracted from a standard, considering only the *base types* of its elements (such as *process area*, *goal*, and *practice* in the case of CMMI-DEV), disregarding any related content (domain-specific notions such as *Requirements Engineering* (a *process area*), *Manage Requirements* (a *goal*) and *Obtain an Understand of Requirements* (a *practice*) in CMMI-DEV).

This is essentially a modeling activity that depends on the correct understanding of the standards' structure. Diverse standards provide a description or a schema of their structure (e.g., CMMI-DEV, ISO/IEC 29110), which can be used as basis. However, the standards are usually described in natural language organized in sections of a document, and additional considerations can affect the modeled structure. In some cases, a generalization of a set of elements is needed to better organize the model. For example, the CMMI-DEV structural elements *Specific Practice* and *Generic Practice* are both composed of *Subpractices* and produce *Work Products*. Thus, the abstract generalization *Practice* can be used to establish a general relation with *Subpractice* and *Work Product*, simplifying the model with no loss. Although this activity deals with structural elements, the standard's content should also be analyzed for identifying elements not explicitly treated by the standard declared structure.

Once the SSMs are available, a new step can be performed, adjusting them according to the initiative scope. It happens by including some elements / relations and disregarding others as discussed in the guidelines below:

- When the scope considers elements that are not in the standard's structure, but present in the standard's text, they should be included in the model. For example, even though ISO/IEC 12207 does not include *stakeholder* as part of its structure, this notion is present in the text and can be considered as an element of the ISO/IEC 12207 SSM, if it is in the scope of the initiative.
- Elements and relations, when are not in the scope, can be discarded from the initiative's structural model. In this case, the discarded elements / relations should be omitted or marked in a different color (e.g. in pink) in the diagram.
- Some structural elements, even not in the scope, can be useful as sources of information. In this case, they are discarded from the SSM, but taken as information sources, from where other considered elements can be extracted. For example, ISO/IEC 29110 has the element *Objective*. If it is not relevant for the scope, the Objectives' text can be used for identifying other elements such as activities and work products.

QA Initiative: 12207 SSM



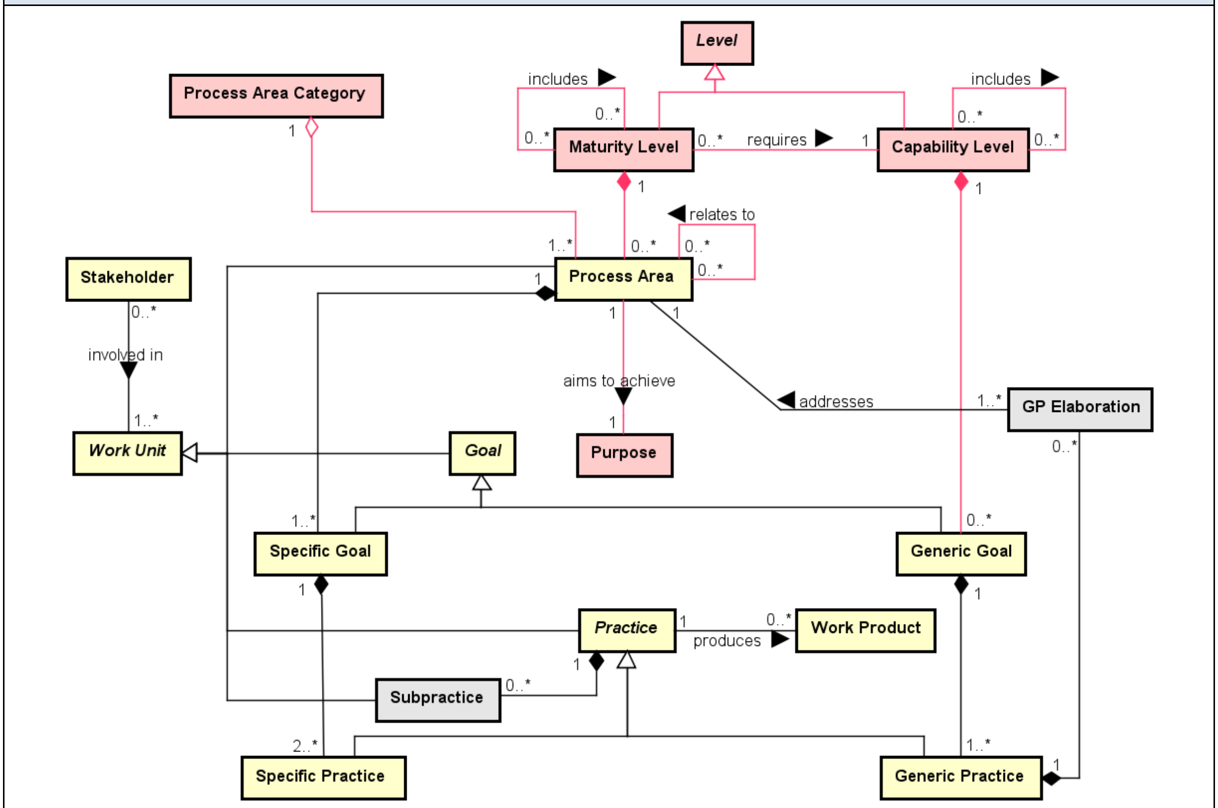
Elements in yellow and relations in black were those considered for being extended in the initiative (according to the defined scope); elements and relations in pink were discarded; elements in gray were used only as information source.

As an example, in the ISO/IEC 12207 SSM (from the QA Initiative), the elements *Process Category*, *Process*, *Activity*, *Task*, *Purpose* and *Outcome* are part of the standard declared structure. *Work Unit* is just a generalization added for organizing the model. *Stakeholder* and *Artifact* are common elements present in the standard's text and thus included in the SSM. In this particular initiative, *Purpose* and *Process Category* and associated relations (in pink) are not part of the scope; and *Outcome* (in gray) will be used only as information source.

Some guidelines are important for this activity: (i) The SSM elements regard the general notions used to describe the standard (e.g. *process*, *work product* etc.), and not a particular section / process. The user should be careful not to include content elements (e.g. the *Design Process*, the *Requirements Document*) in the structural model. All the content elements shall be specialized from the structural elements in the next phase. Other previously defined structural models can be helpful as examples. (ii) The elements to be modeled regard the subject described by the standard, usually in the process's realm (such as *process*, *activity*, *artifact*), and not the standard's description itself (such as section, note, etc.). (iii) During the addition or disposal of model elements, some adjustments may be necessary to the scope. Keep the scope always consistent with the selections done.

Moreover, natural language is not as precise as a model, and some inconsistencies can appear at this moment. For example, ISO/IEC 12207 defines *activity* as a set of *tasks*; however, some *activities* have a single, equivalent, *task* (are they the same?); CMMI-DEV includes *tools* in the typical *work products* list). The creation of the SSMs is fundamental to the approach, since it precisely defines the structure to be mapped to the ontologies and other standards, and to be extended to derive the content models.

QA Initiative: CMMI SSM



A2.2. Map Standards' Structural Models to Core Ontologies

The Structural Vertical Mapping establishes the matches between the standards' structural elements and the SEON View core concepts. Each SSM shall be mapped to the SEON View by comparing the elements of the SSM (only those considered in the scope) with the core concepts of the SEON View.

The result is a table with the core concepts (in the first column) mapped to the corresponding structural elements (in the remaining columns). The QA Initiative box shows the structural mappings of CMMI-DEV and ISO/IEC 12207 to a SEON core ontology fragment. These structural matches link the compatible structural notions, defining all the possible future matches between the content elements and domain concepts. Thus, since *Process Area* (from the CMMI SSM) matches only to *Performed Process* (in the SEON View), during the content mappings, the CMMI *process area* specializations (e.g. *PPQA*, *REQM*) could be matched only to the SEON *performed process* specializations (e.g. *Quality Assurance Process*, *Requirements Management Process*) and not to other concepts, such as *activities* or *artifacts*. Observe that in this activity all the notions are related to the same *ontological type* from UFO (e.g. *event*, *object*, *agent*) for preventing that the structural elements are matched to concepts with distinct nature.

QA Initiative: Structural Mapping		
Core Ontology Concept	**CMMI-DEV Element**	**ISO/IEC 12207 Element**
Performed Process (event)	Process Area (event) Specific Goal (event) Generic Goal (event)	Process (event) Activity (event)
Performed Activity (event)	Specific Goal (event) Generic Goal (event) Specific Practice (event) Generic Practice (event)	Process (event) Activity (event) Task (event)
Artifact (object)	Work Product (object)	Artifact (object)
Stakeholder (agent)	Stakeholder (agent)	Stakeholder (agent)

In this activity, some practical considerations regarding the elements' semantics should be considered:

- Since SEON core ontologies are grounded in UFO, the basic ontological distinctions are essential for supporting the structural mapping. Ideally, an ontological analysis should be conducted for each standard's structural model, grounding them in the same foundations of SEON (i.e., UFO) before doing the comparisons. However, when a complete ontological analysis is considered too costly for the harmonization initiative, a simpler alternative is to identify the basic ontological type of each model element, namely: *event*, *object*, *agent* and *moment*. Once the *ontological type* of an element is identified, during the mapping, each element needs to be compared only with concepts of the same ontological type. For example, given that "work product" is classified as an *object*, it needs to be compared only with notions also classified as *object*, and not with *agents*, *moments* or *events*.
- Divergences between the standards can appear. For example, ISO standards use a very similar classification for work units: *Processes* are composed of *Activities* that are composed of *Tasks*. Each of these definitions is also shared by the standards. However, considering distinct ISO standards, similar work units are classified differently. To illustrate, *Requirements Analysis* and

Architectural / Detailed Design are considered *processes* in ISO/IEC 12207 and *activities* in ISO/IEC 29110. Thus, in this step, the meaning and the application of each element must be analyzed for establishing suitable matches.

- Sometimes, a standard collapses different things into a single element. For example, ISO/IEC 12207 states that an Outcome "*describes one of the following: (i) production of an artefact; (ii) a significant change in state; (iii) meeting of specified constraints*". According to UFO, it can be considered a *situation*. However, situations are complex entities that can encompass *events*, *objects* and other ontological distinctions, being hard to establish a precise comparison. In such cases, it can be disregarded as a relevant element, and used only as a source of information for other elements (from *Outcome*, for example, can be extracted activities and artifacts).
- Pay attention to the element's **meaning**. Consider also its definition, context and relations, and not only the term used to name it.

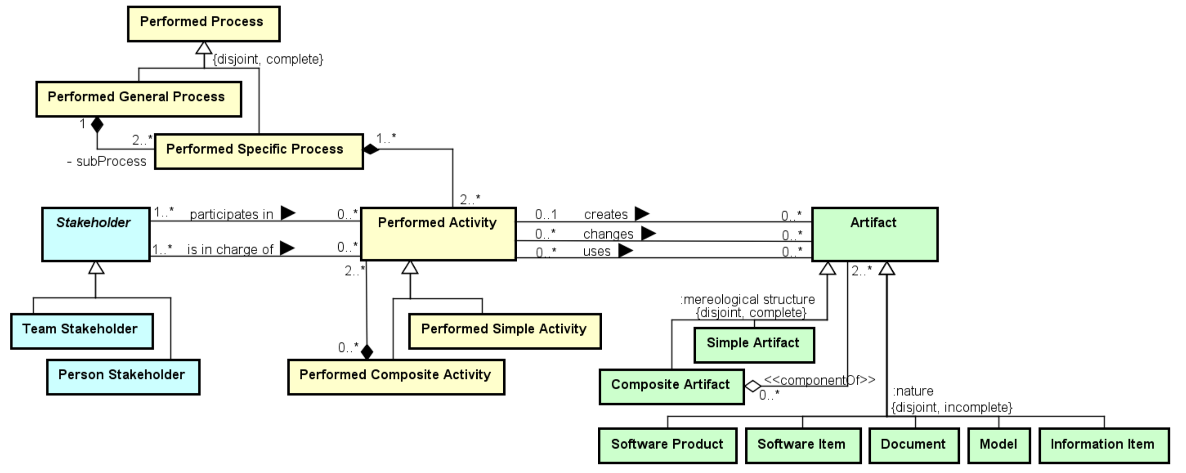
A2.3. Develop the Integrated Structural Model

An Integrated Structural Model (ISM) shall be developed for representing a unified view of the standards' structures. It is created by copying the portion of the SEON View containing the core concepts with matches or that are relevant for the integrated model. Thus, all the previous matches to the core concepts remain valid when they are copied into ISM elements. Additionally, the ISM can be modified by including or adapting elements, relations and definitions based on the previous mappings.

All the SSM elements must have a correspondence in the ISM. Thus, when SSM elements relevant for the initiative remain with no matches, the ISM shall be complemented with new elements. These new elements are created directly in the ISM and mapped to the corresponding SSM elements (those previously with no matches). The mappings between the ISM and the SSMs are said Horizontal Mappings. For illustrating, if the elements "*Result*" and "*Expected Result*" are present in two different SSMs, and no corresponding concept is found in the core ontologies, a new ISM element, e.g. "Result", shall be created and matched to the corresponding ISM elements. This activity produces an integrated model providing a structural consensus of the involved core ontologies and standards' structures that will be used to support the derivation of the Integrated Content Model (ICM) in the next phase.

QA Initiative: Integrated Structural Model (ISM)

Since no modifications were made in the ISM, it corresponds to the core portion of the presented SEON View (the fragment above the dashed line in the SEON View diagram), as shown below.



5.4.3 *Content Harmonization*

The third phase focuses on harmonizing the contents of the involved standards. It aims at identifying the key elements from the standards' contents and mapping them to the ontologies, and creating an integrated model. The main results are the Standards' Content Models (SCMs), their mappings to the domain ontologies, and the Integrated Content Model (ICM). The ICM is Harmony's main result and represents an integrated view of the harmonized standards. Figure 5.7 shows the activity diagram for this phase. Next, each activity is described.

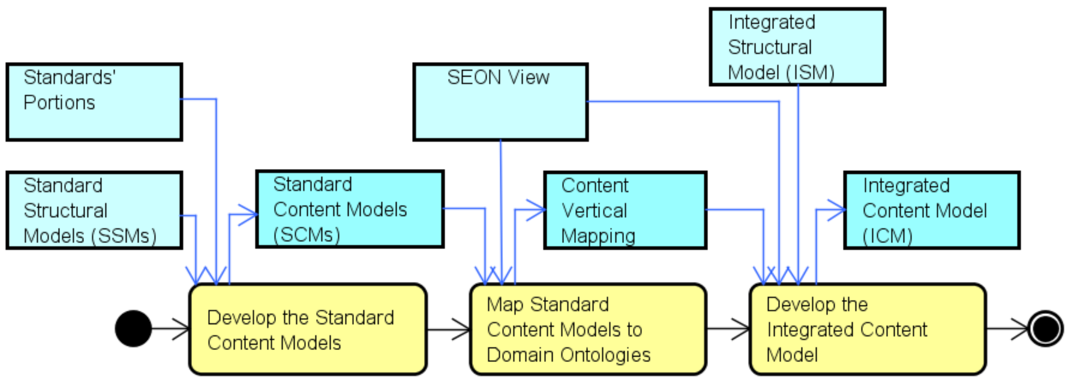


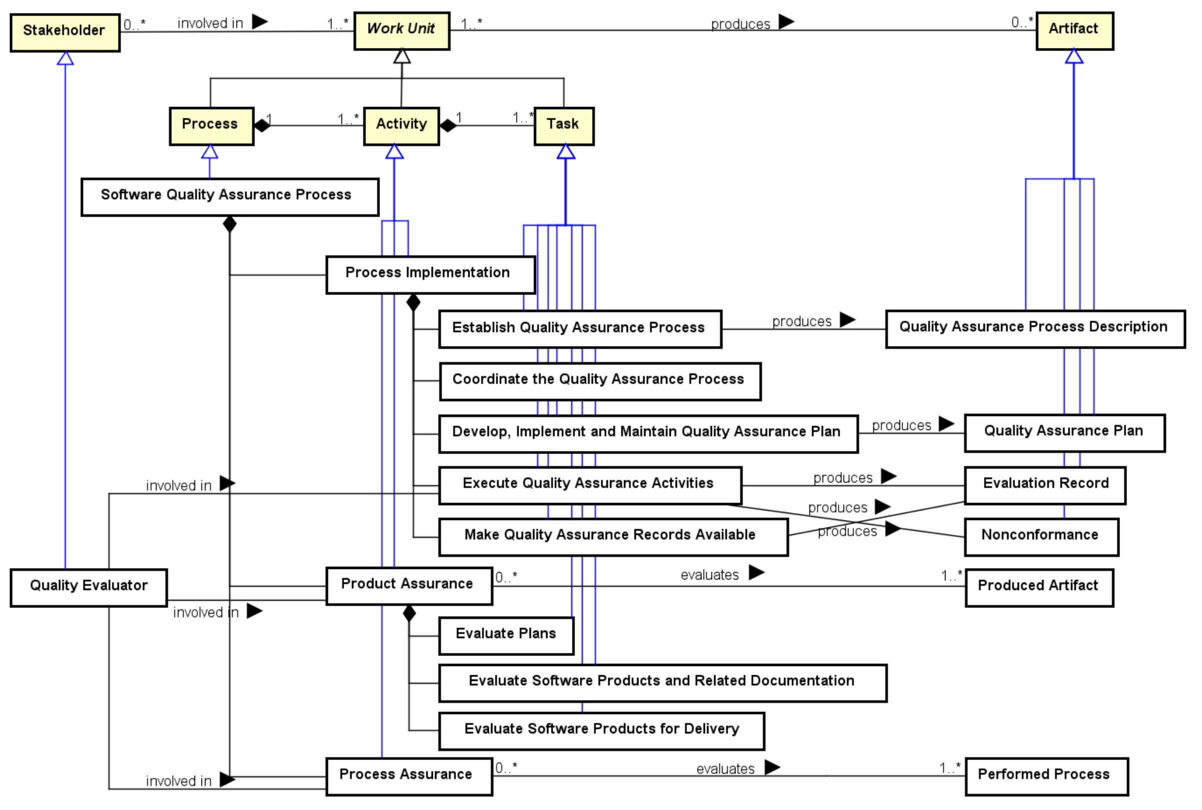
Figure 5.7. Content Harmonization.

A3.1. Develop the Standards' Content Models

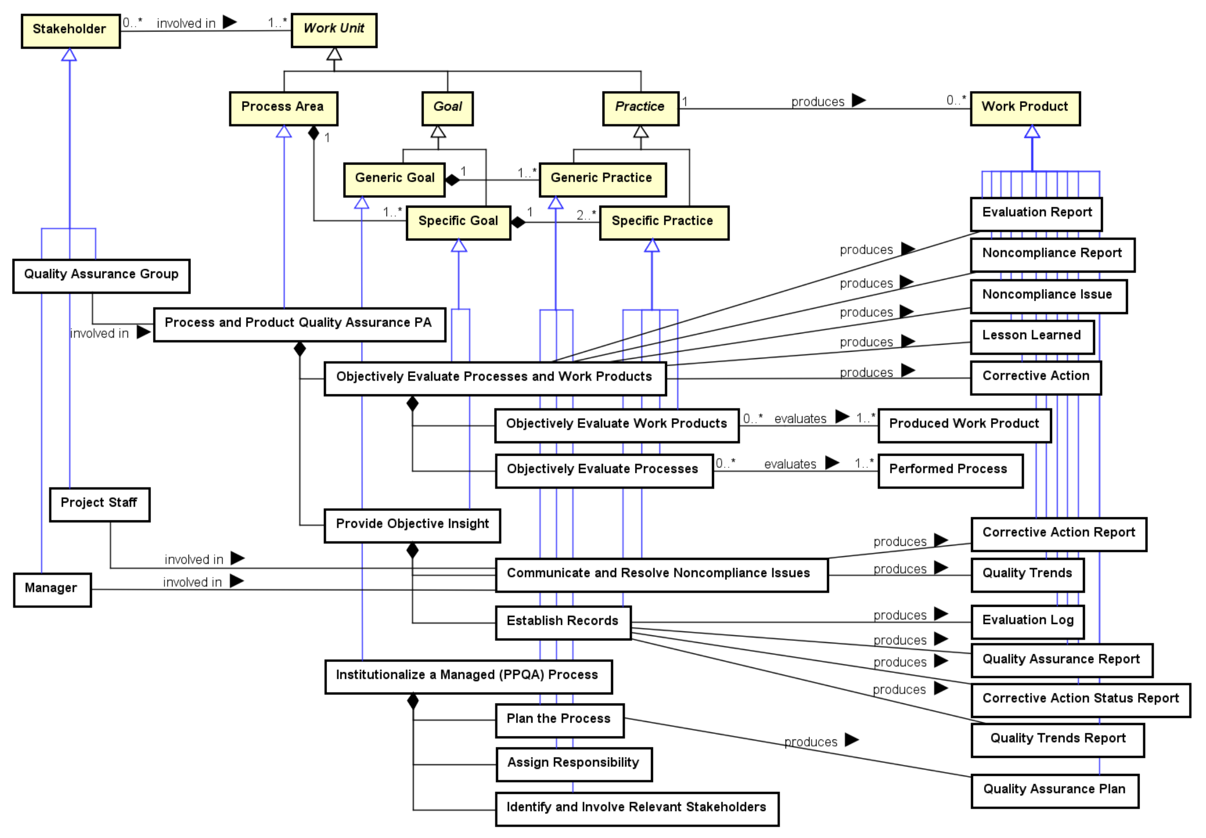
This activity aims to produce the Standards' Content Models (SCMs) by extending the structural models (SSMs) according to the defined scope and information extracted from the standards. A <u>conceptual model</u> shall be built <u>for the content of each standard</u>. The SCM organizes the standard's portions more precisely, representing the relevant elements and the relations between them. The development of the SCMs is essential, since it provides a general view of each involved standard, representing the main information in a precise and concise way, i.e., as a model. It facilitates the analysis and comparison of the standards' pieces, favoring dealing with the standards' semantics and improving the mapping results.

The knowledge extraction is done by analyzing the selected standards' portions and identifying their key elements and relations. The content elements are modeled as specializations of the structural elements (the *base types*) defined in the SSM. For example, in the case of ISO/IEC 12207, its SSM defines the element *Process*, and in its SCM there is the *Software Requirements Analysis Process* element, which is a specialization of the *Process* element. These specializations configure important information in the model, since they set the base type (structural element) of each content element and thus delimit the possible content matches only to those allowed by the structural mappings. The following two boxes show the ISO/IEC 12207 and CMMI-DEV content models (partially).

QA Initiative: 12207 SCM (partial)



QA Initiative: CMMI SCM



This modeling effort is not an easy task, since it aims at creating a model from the standard's natural language text, which can be ambiguous and superficial. Some elements and relations can be hard to identify. Guidelines are given to support this activity:

- It is useful to highlight the elements in the standards' text before including them in the model (or even organizing them in a draft table). Different colors / marks help to identify different types / information (e.g., *activities*, *artifacts*, *stakeholders*, ~~*out of scope*~~). A second (and possibly third) analysis of the standard's text and model is helpful, especially if it is done by another person.
- Try to model only the elements contained in the scope and with types considered in the SSM. When the defined scope includes elements from different parts of the same standard, it could be necessary to include in the model higher-level elements (such as a *process*) out of the scope for matter of organization. In this case, mark the element in red, and discard it during the mappings. Elements partially included in the scope should be modeled and mapped like so.
- Be careful while extracting the standards' information.
  - Seek for implicit elements. Standards are usually described in natural language. Thus, sometimes, important elements are not explicitly defined in the text (especially artifacts and stakeholders). For example, the sentences "*the requirements should be documented*", "*a record for the requirements is created*" and "*the stakeholder shall register the requirements*"

all refer to some kind of *Requirements Document* element. These elements can be found in different parts of the standards' text, including the description of other elements.

- Direct deductions are welcome, but avoid completing the model with information coming exclusively from the user knowledge. For example, if the standard has as activity "*Establish a Project Plan*", it should be modeled the activity *Establish a Project Plan*, and also the artifact *Project Plan* created by the activity (two elements and a relation). However, the model shall be truthful to the standard. If the standard does not mention or clearly suppose an element (or relation), it should not be included in the model, even if it is a missing piece of the resulting model. For example, even knowing the Project Manager is the responsible for establishing a project plan, she/he is not even mentioned in the activity "*Establish a Project Plan*" and should not be modeled.
- After finishing the model, check if it is adherent and complete according to the standard.

- When an element is modeled, set its definition given by the standard (e.g. in the modeling tool). When a definition is not given by the standard, at least describe enough information to understand and compare the element. It will help in the mapping activities.
- Try to keep a similar granularity and abstraction level when modeling the standards. It will be helpful for mapping the elements.

By modeling each SCM from their related SSM the user is creating a model closer to the original standard, more faithful to its contents. It reduces the difficulty involved in the content mapping, since fewer decisions have to be taken together (separation of concerns). First the user models the elements, specializing them from the SSM ones. Then, in the next activity, the user matches each element to the domain ontologies' concepts. Additionally, the standards' models are influenced only by the scope, which makes them more reusable in other initiatives.

A3.2. Map Standards' Content Models to Domain Ontologies

This activity is a Vertical Mapping, attaching the standards' contents to a semantic domain reference. The user shall decide how each SCM element matches to the SEON View's concepts. For each standard's element, the user shall select the concept that better matches it and set a match type.

A match is a relation between notions of two distinct models describing how an element relates to a concept. Just saying that two notions match or not is not enough to properly describe the complex relations between the information underlying these notions. The standards are described in many different ways, varying in organization, structure, adopted language, abstraction level and perspective. Each modeled element captures a piece of information that should be analyzed and linked to another related piece of information in another model. The matches must describe more precisely what these links represent. For instance, two notions can be equivalent, a specialization / generalization, or related by a part of relation, among others. In our solution, a match involves, besides the pair of notions, a type, a coverage range, and additional comments. Table 5.1 presents the nine types of matches Harmony provides.

Table 5.1. Harmony Match Types.

MATCH TYPE	SYMBOL	MEANING	EXAMPLE
EQUIVALENT	**A [E] B**	A is Equivalent to B. Element A represents a notion that **is equivalent to** the notion represented by Concept B.	(Element) Risk Plan **[E]** (Concept) Plan of Risks
PART OF	**A [P] B**	A is Part of B. Element A covers a notion that **is part of** the notion represented by Concept B (B includes A).	(Element) Risk Plan **[P]** (Concept) Project Plan
WIDER	**A [W] B**	A is Wider than B. Element A covers a notion that **is wider than** the notion represented by Concept B (A includes B).	(Element) Risk Plan **[W]** (Concept) Mitigation Plan **{contingency actions not covered}**
OVERLAP	**A [O] B**	A Overlaps B. Element A covers a notion that **overlaps** the notion represented by Concept B (A and B include the same part P).	(Element) Requirements Verification and Validation **[O]** (Concept) Requirements Validation and Agreement **{verification not covered}**
SPECIALIZATION OF	**A [S] B**	A is a Specialization of B. Element A represents a notion that **specializes** the notion represented by Concept B.	(Element) Software Designer **[S]** (Concept) Developer
GENERALIZATION OF	**A [G] B**	A is a Generalization of B. Element A represents a notion that **generalizes** the notion represented by Concept B.	(Element) Requirement **[G]** (Concept) Functional Requirement
ACTS AS	**A [A] B**	A Acts as B. Element A represents a notion that can **act as** the *role* represented by Concept B.	(Element) System Analyst **[A]** (Concept) Requirements Reviewer (a System Analyst can play the role of Requirements Reviewer)
IS ACTED BY	**A [B] B**	A is acted By B. Element A represents the notion of a *role* that can be **acted by** the notion represented by the Concept B.	(Element) Requirements Agreement **[B]** (Concept) Client E-mail (a Client E-mail can play the role of Requirements Agreement)
NO RELATION	**A [-]**	A has no relation. Element A represents a notion that **has no corresponding relation** with any concept in the ontology.	(Element) Sequence Diagram **[-]** (there is no corresponding concept in the ontology)

An important consideration in a mapping is which is its coverage, i.e., how much the source model (SCM) is covered by the target model (the SEON View). The standard's coverage is given by the sum of the coverage of its elements. Elements with an Equivalent [E] or Part of [P] matches are considered **fully covered**. Elements with No Relation [-] are considered **not covered**. Elements matched using any of the other types are considered **partially covered**, **largely covered**, or **fully covered**, depending on the situation. Moreover, when the same element has multiple matches (e.g., with Wider [W] and/or Overlap [O]), it can be set as **fully covered**, by a *composite match* (a match grouping two or more previous matches for assuring the full coverage of a source element).

For the matches that only partially cover an element (Wider [W] and Overlap [O], for example), a comment shall be included {in brackets}, detaching the non-covered part(s). For assuring the proper coverage, not **fully covered** elements should be treated in the next activity. For example, consider the element *Product and Process Evaluation*. If the SEON View comprises only the evaluation of processes, a Wider [W] match is created with a comment such as "{Product evaluation not covered}". Once the element is not fully covered (only a portion of it is in the SEON View), the remaining portion is to be dealt while building the ICM (next activity).

We provide the following guidelines for performing this activity:

- Compare the standards' elements with the domain concepts considering the structural mappings, since the possible matches are expressed there. For example, the CMMI SSM has a *Work Product* element, which is mapped to (*corresponds to*) *Artifact* in SEON. Thus, during content mapping, *Work Products* specializations should be mapped only to *Artifacts* specializations.
- When the SSM has an element out of scope (represented for organizing the model, or for completing a process), it shall be discarded during the mapping. However, sometimes an element is partially out of scope. For example, if the harmonization initiative scope is "Project Planning", and the standard has an activity *Plan and Monitor the Risks,* it should be modeled (due to the *plan* part). In this case, the matches must consider always the entire element, i.e., if the ontology has concepts for the activity *Risk Planning* and the process *Project Planning*, that element (*Plan and Monitor the Risks*) is Wider than the first and Overlaps with the second (*Plan and Monitor the Risks [W] Risk Planning; Plan and Monitor the Risks [O] Project Planning*). This approach preserves the match for possible reuse in other initiatives.
- Focus on the **meanings** of each element and concept, instead of on the term used to name them. Definitions and relations between elements / concepts can help. The standard's text may also be consulted. It is important to stress that the matches are established based on the <u>meaning of each individual notion</u>. For instance, if an element representing a process has the same meaning of a concept representing a process, they are matched as Equivalent. However, it does not mean that all their parts (activities and tasks) must also be Equivalent or fully covered. Two processes can have the same intentions but not describe the exactly same parts.

- Try stronger matches first. If an element has an equivalence [E] match, it shall be set; otherwise, composition, specialization and role-related matches can be attempted. Once an element is **fully covered** (with an equivalence, part of, or other matches that together cover all its extension) other matches become irrelevant and do not need to be expressed.
- This activity is fully supported by the *Mapper* mapping tool, assuring consistency and presenting the results. If the tool is not used, tables should be built with the following columns: element, match type, comment, coverage and concept.

QA Initiative: CMMI Vertical Mapping (partial)

Background colors in the first column points out the elements' coverage: green for fully covered; yellow for partially covered; blank for not covered. Composite matches are shown with the target (third column) including two or more concepts.

CMMI Element	Match	SEON View Concept
EVENTS		
Process and Product Quality Assurance PA	EQUIVALENT	Quality Assurance Process
Objectively Evaluate Processes and Work Products	EQUIVALENT	Adherence Evaluation
Objectively Evaluate Work Products	WIDER {*Noncompliance identification not covered*}	Artifact Evaluation
	WIDER {*Artifact evaluation not covered*}	Noncompliance Identification
	EQUIVALENT	(Artifact Evaluation + Noncompliance Identification)
Performed Process	ACTS AS	Evaluated Process
Objectively Evaluate Processes	WIDER {*Noncompliance identification not covered*}	Process Evaluation
	WIDER {*Process evaluation not covered*}	Noncompliance Identification
	EQUIVALENT	(Process Evaluation + Noncompliance Identification)
Provide Objective Insight	WIDER {*Communication not covered*}	Noncompliance Control
Communicate and Resolve Noncompliance Issues	WIDER {*Communication not covered*}	Noncompliance Resolution
	WIDER {*Communication and resolution actions not covered*}	Noncompliance Closing
Establish Records	NORELATION	
...		
OBJECTS		
Evaluation Report	EQUIVALENT	Evaluation Report
Noncompliance Report	NORELATION	
Noncompliance Issue	EQUIVALENT	Noncompliance Register
Corrective Action	EQUIVALENT	Corrective Action Register
Produced Work Product	ACT AS	Evaluated Artifact
Quality Assurance Report	NORELATION	
Corrective Action Status Report	NORELATION	
Quality Assurance Plan	EQUIVALENT	Quality Assurance Plan
...		

AGENTS		
Quality Assurance Group	WIDER {*Documentation and reporting responsibilities not covered*}	Quality Auditor
Project Staff	EQUIVALENT	Project Team
Manager	EQUIVALENT	Project Manager
Mapping Analysis: the CMMI-DEV PPQA process area is covered in 61% by the SEON View. Most of the uncovered portion regards documentation and communication activities and the related artifacts and stakeholders.		

In the following, we present some extra information that is not present as part of the provided documentation of Harmony, however it is relevant for a further discussion in this text.

The **match types** refer to the modes a source element (from a Standard) associates to a target concept (from an Ontology). A match can be seen as a binary relation linking two notions of different models (the SCM and the SEON View). As explained, the matches are allowed only between notions with the same *ontological type*. More than that, content matches shall be made according to the structural matches (*base type* compatibility). Each match type carries a set of information regarding its meaning, applicability, properties, and coverage range. The meaning explains what the match type is. The applicability refers to which kinds of notions (regarding the *ontological type*) each match type applies. The properties regarding symmetry and transitivity are described and, for asymmetric relations, the inverse relation (another match type) is pointed out. These properties are relevant for consistency check and match deductions. Finally, the possible coverage ranges are defined. The coverage describes in which extension a source element is covered in a match[10]. This is useful for calculating how much a model is covered by another model and to identify the uncovered parts. In the following, the nine match types in the Table 5.1 are presented in four groups and further discussed.

Comparison Matches (**Equivalent** and **No relation**) are the most basic match types ("*match* or *do not match*") and can be established for any allowed combination of notions. Thus, they can be applied for any pair of notions with compatible types (e.g. activities (*event*), stakeholders (*agent*) or artifacts (*object*)).

- **Equivalent [E]** is used when two notions represent the same information in both models, having the same meaning and characteristics. Equivalent is a symmetric relation, i.e., if the source notion A is equivalent to the target notion B, then B is also equivalent to A (A [E] B $\rightarrow$ B [E] A). It is also a transitive relation (A [E] B $\wedge$ B [E] C $\rightarrow$ A [E] C). These properties are useful for comparing different standards already mapped to the same ontology and to identify possible inconsistences (in the mappings and in the standards). The Equivalent match implies in a complete coverage, i.e., A [E] B means A is *fully covered* by B.

[10] The coverage ranges are used in the Mapper tool in the following extension: *not covered* (0%), *partially covered* (33%), *largely covered* (67%) and *fully covered* (100%).

- **No relation [-]** applies when a given notion has no corresponding notion in the target model. Differently from the other match types, it is not a binary relation and just set the source element as *not covered*.

Composition Matches (**Part of**, **Wider** and **Overlap**) are devoted to matches involving complex notions which can be decomposed (have identifiable parts), such as *events*, *complex objects* and *collective agents*. It is natural to say that an activity (*event*) is wider than another activity (*event*), or that a stakeholder (*agent*) is part of a team (*collective agent*); however, it is not reasonable to say that a stakeholder (*agent*) is part of another stakeholder (*agent*). As examples, *WBS [P] Project Plan*; *Project Planning [W] Resources Planning*; *Software Architect [P] Project Team; Establish the Use Cases Specification [O] Develop the Use Cases and Classes Models*.

- **Part of [P]** is applied when the source element represents a part of the target concept. If A [P] B, then A represents a specific portion of B, that is narrow than B. Part of is a transitive (A [P] B $\wedge$ B [P] C $\rightarrow$ A [P] C) and asymmetric relation. Its inverse relation is the Wider [W] match type (A [P] B $\rightarrow$ B [W] A). Moreover, A [P] B implies in a complete coverage, i.e., A is *fully covered* by B.
- **Wider [W]** is applied when the source element represents more information than the target concept. If A [W] B, then A represents B and something more. Wider is also a transitive (A [W] B $\wedge$ B [W] C $\rightarrow$ A [W] C) and asymmetric relation, and its inverse is Part of [P]. Moreover, A [W] B implies in an incomplete coverage, having the user to define, according to the notions' correlation, if A is *partially covered* or *largely covered* by B.
- **Overlap [O]** applies when the source element has a part in common with the target concept. If A [O] B, then there is an implicit notion C where C [P] A and C [P] B. Overlap is a symmetric relation (A [O] B $\rightarrow$ B [O] A) and implies in an incomplete coverage (the source element is *partially* or *largely covered*).

For aiding users to understand and apply the equivalence and composition matches, an analogy to the set theory can be used. Although it is not intended to apply the set theory foundations here, a similar representation is useful to further describe these match types. As Figure 5.8 shows, an element E is **Equivalent** to a concept C when both notions represent the same information (cover the same portion of the domain). An element E is **Part of** a concept C when E represents a part of C (E is included in C). An element E is **Wider** than a concept C when E represents more than C (C is included in E). Finally, an **Overlap** happens when E and C have a portion in common.

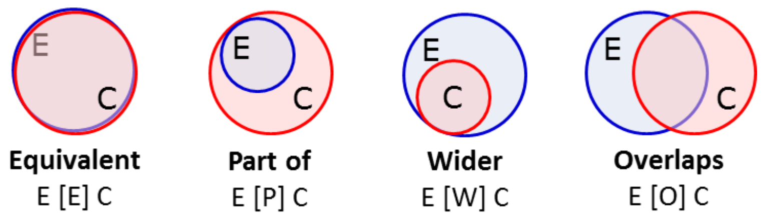


Figure 5.8. Composite Match Types Representation.

Specialization / Generalization Matches (**Specialization** and **Generalization**) can be applied for any notion, but they are more common with *objects* and *agents*. For example, a stakeholder (*agent*) can specialize another stakeholder (*agent*), and an artifact (*object*) can generalize another.

- **Specialization of [S]** is used when the source element represents a specialization of the target concept. Specialization of is an asymmetric and transitive relation and its inverse is the Generalization of [G] match type. It can incompletely or completely cover the source element. For instance, consider the match *ER Model [S] Conceptual Model*. If *Conceptual Model* fulfills the standard's needs, being able to replace the *ER Model* in the SCM, it is a complete coverage. Otherwise, incomplete coverage. Since in specializations the source is more specific than the target, usually specializations are completely covered.
- **Generalization of [G]** is applied when the source element represents a generalization of the target concept. It is an asymmetric and transitive relation (inverse of **Specialization of** [S]). Generalization matches can incompletely or completely cover the source element. For instance, consider the match *Developer [G] Programmer*. If *Programmer* fulfills all the model needs and could replace *Developer* in the source model, it is a complete coverage. Since in generalizations the source is more general than the target, usually generalizations are incompletely covered.

Role-related Matches (**Acts as** and **is acted By**) apply when one of the notions is a *role* played by other. This type of match usually occurs between *objects* or *agents*, i.e. an artifact (*object*) can play the role of another artifact (*object*) or a stakeholder (*agent*) can play the role of another one.

- **Acts as [A]** is used when the source element can play the role of the target concept, i.e., the source can replace the target as needed. E.g., *Class Model [A] Design Input*.
- **is acted By [B]** match is the inverse of **Acts as** [A], and applies when the source element is a role that can be played by the target concept. E.g., *Change Requester [B] Project Manager*.

These two types of matches are considered asymmetric and non-transitive. Both can cover the source element completely (when the target notion can replace the source one with no loss) or incompletely (otherwise).

Concerning the mapping consistency, some discussion is deserved. During the mapping, as the matches are being established, some situations should be observed and treated as necessary. These situations refer to restrictions and unusual circumstances that shall be solved to keep the mapping integrity.

As discussed before, independently of the match type, two notions only can be matched if they have the same nature (*ontological type*) and are specialized from compatible notions, i.e., notions that were matched during the structural mapping. For example, a particular *task* from a SCM can be matched to a particular *activity* from an ontology (since both are *events*), but only if in the structural mapping the base element *Task* is matched to the core concept *Activity*.

The remaining situations regard multiple matches considering the same notion (in the source model or in the target ontology). For instance, the same element can be matched to multiple concepts (two or more) by means of any match type (e.g., **A** [*] B1 and **A** [*] B2). However, when an element is already *fully covered* (e.g. by an Equivalent [E] or Part of [P] match), a new match is not allowed. When an element is *partially / largely covered* in multiple matches, the user shall inform if the sum of the matches *fully covers* the element, resulting in a *composite match*. For example, considering the matches A [W] B1 and A [W] B2 (both have incomplete coverage), the user should inform if these two matches together configure a composite match, i.e. if it is the case that A is equivalent to the sum of B1 and B2 (A [E] (B1+B2)) or not. For example, if *Plan and Monitor Risks [W] Plan Risks* and *Plan and Monitor Risks [W] Monitor Risks*, then a composite match can be created for defining that *Plan and Monitor Risks [E] (Plan Risks + Monitor Risks)*, *fully covering* the source element.

In the opposite way, another situation occurs when multiple source elements (two or more) are matched to the same target concept (e.g., A1 [*] **B** and A2 [*] **B**). If one (or more) of the matches *fully covers* one of the elements, this can indicate some overlap in the standard's model, i.e., the same information is modeled in distinct elements. This situation points out a modeling error (when building the SCM) or an inconsistency in the standard. The last case (inconsistency in the standard) often happens in ISO/IEC 12207 when it defines an activity as a set of tasks containing just one task. The activity and the task have the same meaning, being matched to the same concept (a standard's overlap). For example, the *Software Configuration Management Process* is composed of six activities, and each activity is composed of a single task. For instance, the activity *Configuration Identification* (7.2.2.3.2) is described as "*this activity consists of the following task:* (7.2.2.3.2.1) *A scheme shall be established for the identification (...)*". Since activity and task represent the same thing, they should have exactly the same matches.

The properties, restrictions and solutions described above are automated in the Mapper tool for supporting the user through the process and keeping the integrity during the harmonization effort.

For each mapping produced, an analysis shall be done for summarizing and highlighting the main coverage aspects. This analysis shall objectively point out the coverage achieved (%), the main portions remaining uncovered in the base standard, and other relevant information identified during the mapping.

Finally, it is important to highlight that the goal is not to match every element achieving the maximum coverage (100%). The ontology can simply not cover some aspects of the standards. Ontologies focus on representing the domain; while standards, besides the domain, are usually concerned in put in evidence aspects such as documentation and verification (specially the certification standards). Thus, it is usual to find some artifacts, activities and other elements regarding such aspects that are not completely covered by the ontologies. This is not a problem at all, since the focus in this activity is on creating as many suitable matches as possible.

A3.3. Develop the Integrated Content Model

The Integrated Content Model (ICM) is a consensual model representing a harmonized view of the involved standards' portions. It is the main result of applying Harmony. Analogously to the ISM, it is built by copying the domain concepts from the SEON View, making them ICM elements. Then, the ICM can be modified by adding / changing / removing elements, relations and definitions focusing on representing a harmonized model and maximizing the standards' coverage.

All SCM elements should be covered by the ICM. Thus, when SCM elements relevant for the initiative remain not fully covered in the vertical mapping, the ICM shall be complemented with new elements for dealing with these uncovered portions. The new elements are incorporated to the ICM model and matched with their related SCM elements (those previously uncovered). The match types and rules in this Horizontal Mapping are the same presented for the vertical mappings. It is important to observe that an ICM element can have matches with several elements even from distinct standards. The intention is to identify all the uncovered portions and solve them. For example, in a Project Management harmonization initiative, if the artifacts *Stakeholder Allocation* and *Stakeholder List*, from the same or distinct standards, remained uncovered, they can raise the new harmonized artifact *Stakeholder Allocation Plan* in the ICM, with the following matches: *Stakeholder Allocation [E] Stakeholder Allocation Plan*; and *Stakeholder List [P] Stakeholder Allocation Plan*.

When a new ICM element is created, it shall be added to the model, setting its *base type* (an ISM element), relations to the existing elements, and definition. Additionally, elements and relations previously defined in the SEON View can be modified (names, definitions and cardinalities, for example) or even excluded, if no relevant correspondence is found in any standard. Since the tasks for creating ICM elements (and related matches) and accommodating then in the model support each other, they shall be done simultaneously.

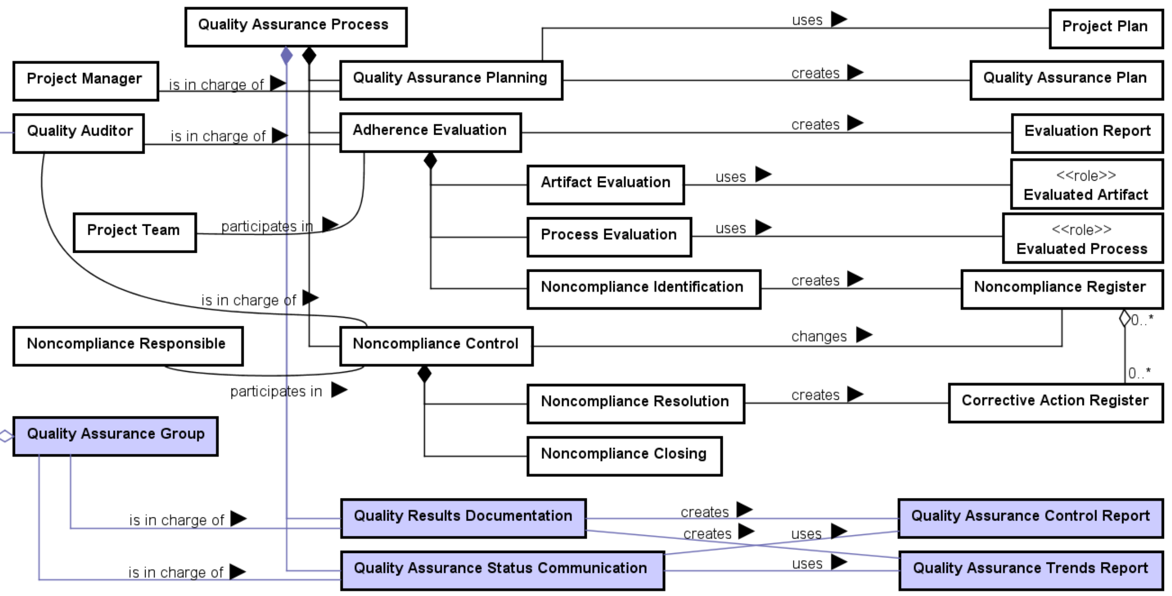
The following guidelines apply for this activity:

- The main criterion for creating new elements is relevance for the initiative scope. If an element appears only in one standard and it is relevant for the defined scope, it should raise a new ICM element. On the other hand, similar elements appearing in many standards, but not relevant for composing the harmonized view (they are already represented as part of other elements or are not important for the scope), could be discarded. For example, standards usually present a set of possible artifacts as activity outputs. If the main artifacts, representing the relevant activity output, are already covered, the other options lose importance and can be disregarded.
- New elements complement the ICM. Thus, seek for representing them in harmonious way with the previous model. Try to use granularity and abstraction levels similar to the SEON View concepts. Assembling, fragmentation, generalization and specialization of elements can be useful.
- This activity is supported by the *Mapper* tool plus a modeling tool. If the mapping tool is not used, a table should be built for presenting the results, with the following columns: elements, match types, ICM element, and definition. Justifications can be textual.

After building the ICM, with the correspondent mappings, the standards' coverage rises to something next to 100%. When the complete coverage is not achieved, the remaining elements shall be analyzed, justifying the reasons for that (usually elements only partially in the scope or indirectly covered).

QA Initiative: Integrated Content Model (ICM)

QA ICM built from a copy of the SEON View domain fragment (in blank) plus the new elements created for the ICM (in purple).



Considering the ICM, the standards' coverage achieved 95% for CMMI-DEV SCM and 92% for ISO/IEC 12207 SCM (against 61% and 91%, respectively, in the vertical mappings). Six elements were not fully covered because they are more general, being only partially in the initiative scope. For example, *Identify and Involve Relevant Stakeholders* from CMMI-DEV and *Software Problem Resolution Process* from ISO/IEC 12207 are not specific for the Quality Assurance realm. Thus, the produced Quality Assurance ICM is produced to cover only the part considered in the initiative scope.

The new elements' descriptions and matches as well as the elements' justifications are in the complete initiative report (Appendix A).

Considering the initiative scope, the ICM achieved 100% coverage of both models, thus we consider it a unified model of CMMI-DEV and ISO/IEC 12207 regarding Quality Assurance processes.

We should have in mind that, the ICM is not an ontology, but a conceptual model representing the standards' consensus. Since most of its elements are copied from the SEON View, all the matches established (including those from the vertical mappings) have ICM elements as targets. The model *per se* is a valuable artifact. Furthermore, it can be used for diverse purposes: (i) being reused in a new initiative for harmonizing an additional standard; (ii) supporting the direct mapping between the involved standards; (iii) providing a knowledge reference for SPI initiatives in multimodel environments; (iv) being the basis for defining processes; and (v) providing improvement suggestions for enhancing SEON.

At the end of the process, we should produce a report containing the structural and content standards' models (SSMs and SCMs), the integrated models (ISM and ICM) and the respective mappings of the standards to them. Appendix A provides the full report for the Quality Assurance Initiative.

5.4.4 Going Beyond

The harmonization initiative results can be useful for other purposes, such as SEON evolution, process definition, and comparisons between pairs of standards, as well as for other harmonization initiatives. Thus, other activities can be performed after applying Harmony.

The integrated models (ISM and ICM) are valuable sources for evolving SEON. The new elements and relations incorporated to these models represent consensual or relevant information extracted from diverse standards that is not present in the SEON View. Although the standards sometimes present some inconsistences, as argued, they are indubitably useful sources of mature and consensual knowledge. Thus, the new identified elements can be useful to perform an analysis for evolving SEON core and domain ontologies.

The ICM, together with the matches with the involved standards, captures a unified view of multiple standards that can be used as basis for defining software processes. The ICM, as a model, provides the main process elements (process, activities, artifacts, stakeholders and so on), their relations, descriptions, and the links (matches) to standards' sources. It constitutes a valued material for starting a process definition activity. Such activity can be performed by using the ICM as basis for defining the process structure and main assets. However, process definition involves additional information not present in the standards. Aspects related to the organizational culture and objectives should be taken into account. Moreover, an effective software process needs to be completed with supplementary information, such as more detailed descriptions, sequencing, inputs and restrictions, which is rarely provided by standards.

The ICM can also be useful for establishing direct mappings between pairs of standards. Once all involved standards are mapped to a common reference (the ICM), the mapping between any pair of these standards can be facilitated. Direct mappings are useful in contexts where the similarities and differences of a pair of standards should be made explicit, such as for organizations which adopted a standard and plan to implement another; and for standards providing alignments to other standards (e.g. MR-MPS-SW and ISO/IEC 29110). This activity is very similar to the Vertical Mapping, using the same match types (see Table 5.1) and similar rules, however, mapping a source standard to a target standard (and not a target ontology). For example, consider an initiative harmonizing portions of ISO/IEC 12207 and CMMI-DEV related to Software Measurement that has produced a 12207 SCM (representing the 6.3.7 Measurement Process) and a CMMI SCM (representing the Measurement and Analysis PA), both mapped to the resulting ICM. These results can support the establishment of a direct mapping between the Measurement Process (from ISO/IEC 12207) and the Measurement and Analysis PA (from CMMI-DEV). More than that, applying the match types properties, several matches of the direct mapping can be even deduced from the vertical mappings.

Moreover, a direct mapping allows for a closer analysis of a pair of standards, since it provides information on how the elements of both standards are related. With this mapping, we can understand the coverage of a standard over another. While the mappings from the standards' models to the ICM offer an integrated view of the involved standards, the direct mappings provide a valuable support for analyzing their similarities and differences. In a standard-to-standard mapping, most of the times the complete coverage is not possible. Due to the standard's scope and characteristics, usually a standard's portion covers only partially another standard's portion (a standard A portion can cover 50% of a standard B portion, and another B' portion can cover only 30% of A' portion).

Although this activity is not effectively included in Harmony, we have further explored it and included in the *Mapper* tool. Thus, it deserves a further explanation on its rationale. The mapping is performed very similarly to the vertical mapping; however, after the previous mappings, much information can be reused. With all involved standards properly mapped to the ICM, the matches already established can be used for deducing some matches for a pair of standards. For example, in an initiative involving standards StdA and StdB, if an element A (from StdA) is Equivalent to an element C (from the ICM), and an element B (from StdB) is also Equivalent to C, since equivalence is a transitive relation, we can assert that A is Equivalent to B (A [E] C $\wedge$ B [E] C $\rightarrow$ A [E] B).

Table 5.2 presents all possible deductions considered for this activity. The first column shows the possible deductions from the combination of two matches with varied types. The premise column illustrates two matches from elements of distinct standards (A from StdA, B from StdB) to the same element (C from ICM). The first result is obtained by directly applying the deduction; the second one is the inverse, obtained by swapping the match types factors. Both results are important, since a mapping is viewed from a source model to a target model. Thus, when the mapping from StdA to StdB is established, the opposite mapping (from StdB to StdA) can also be defined using the inverse relations (Result 2).

Table 5.2. Deductions applied in the Standards' Mapping.

Deduction	Premise	Result 1 (StdA→StdB)	Result 2 (StdB→StdA)
[E]+[E] → [E]	A [E] C ^ B [E] C	A [E] B	B [E] A
[P]+[E] → [P]	A [P] C ^ B [E] C	A [P] B	B [W] A
[W]+[E] → [W]	A [W] C ^ B [E] C	A [W] B	B [P] A
[O]+[E] → [O]	A [O] C ^ B [E] C	A [O] B	B [O] A
[S]+[E] → [S]	A [S] C ^ B [E] C	A [S] B	B [G] A
[G]+[E] → [G]	A [G] C ^ B [E] C	A [G] B	B [S] A
[A]+[E] → [A]	A [A] C ^ B [E] C	A [A] B	B [B] A
[B]+[E] → [B]	A [B] C ^ B [E] C	A [B] B	B [A] A
[P]+[W] → [P]	A [P] C ^ B [W] C	A [P] B	B [W] A
[S]+[G] → [S]	A [S] C ^ B [G] C	A [S] B	B [G] A

Proceeding with the mapping, besides the deductions provided from the previous information, new matches should be added according to the standard's elements remaining uncovered. It is done until all the suitable matches are established in each desired combination of standards. This activity can produce up to two mappings for each pair of involved standards. For example, in an initiative involving standards StdA, StdB and StdC, six direct mappings are possible (StdA→StdB, StdB→StdA; StdA→StdC, StdC→StdA; StdB→StdC, and StdC→StdB), but it can be planned to produce only the two first, for example.

In the Quality Assurance Initiative, we have conducted an additional activity for directly mapping the models of the two involved standards (CMMI-DEV and ISO/IEC 12207). It was supported by the Mapper tool, which deduced several matches from the previous mappings to the ICM. Other new more specific matches completed this mapping. The main results are described in the following and detailed in Appendix A.

- ISO/IEC 12207 SCM covers 70% of the CMMI SCM. The uncovered portion regards mostly objective evaluation restrictions and further details in the CMMI-DEV specification for stakeholders and work products.
- CMMI SCM covers 91% of the ISO/IEC 12207 SCM. The uncovered portion regards ISO elements that are only partially in the scope (such as Problem Resolution, that is partially covered by Noncompliance control, but the other types of problems solved are not in the scope of this initiative). Considering the Quality Assurance initiative scope, no relevant portion remained uncovered.

It is important to highlight that these numbers are not precise and just provide a general view of the coverage. The calculations are done considering the sum of coverage of the elements present in the SCMs and each element has the same weight.

5.5 Mapper: the Harmony Supporting Tool

The Harmony Process involves some activities of complex and laborious nature. The user needs to analyze the standards, extract knowledge, model it, and make a lot of decisions on whether and how each notion is related to others. Handling this amount of information can be a wearing and error prone activity, if it is not supported by proper tools. The modeling activities are more general and can be supported by a variety of modeling tools able to produce UML class diagrams. In this thesis, for example, we have used Astah[11] as modeling tool. However, the mapping activities are much more specific and interconnected to our process, requiring a particular solution. In the first attempts, we tried to use some tables for organizing all the elements, concepts and their matches. However, the user overhead for checking the allowed actions and the consistency of the mappings becomes too high. Thus, we have developed a specific software tool for supporting some Harmony activities, called Mapper.

Mapper is a web tool for managing and supporting harmonization initiatives conducted by using Harmony. Its main purpose is to speed up the process and assure information integrity, while the user can focus on the decisions involving the standards' and domain knowledge. The aforementioned Harmony models, rules, match types, properties and restrictions are implemented by Mapper. This section presents the tool's main features and how they support the corresponding Harmony activities.

The tool focuses on the content mapping and integration activities (phase 3), offering for the user a visual and operational environment for establishing consistent matches making use of all information available. The other phases are also contemplated by Mapper, but with a more basic support. The modeling tool is an important ally for producing the involved models.

For creating an initiative in Mapper, some structural parameters need to be provided. The models produced in the modeling tool are imported and made available for the harmonization efforts. Once an initiative is started, a menu page provides seven main actions as shown in Figure 5.9 and described in the following.

[11] http://astah.net/

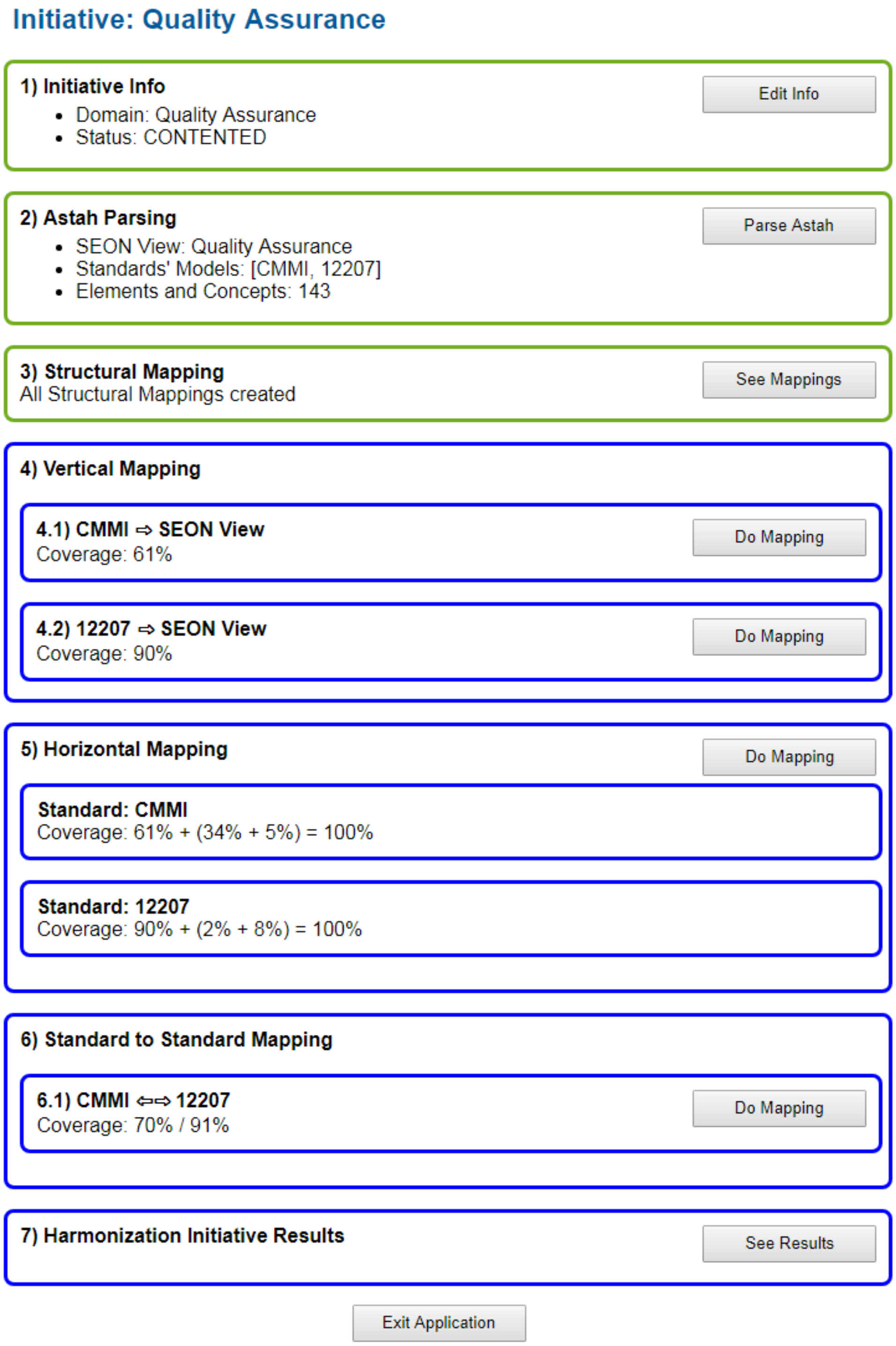


Figure 5.9. Mapper: Main Menu.

(1) Initiative Info

This is a simple action used only for providing the initiative purpose, scope and people involved, as Figure 5.10 shows. It just collects the information resulted from Harmony activities A1.1 (Identify the Harmonization Purpose) and A1.2 (Define the Harmonization Scope).

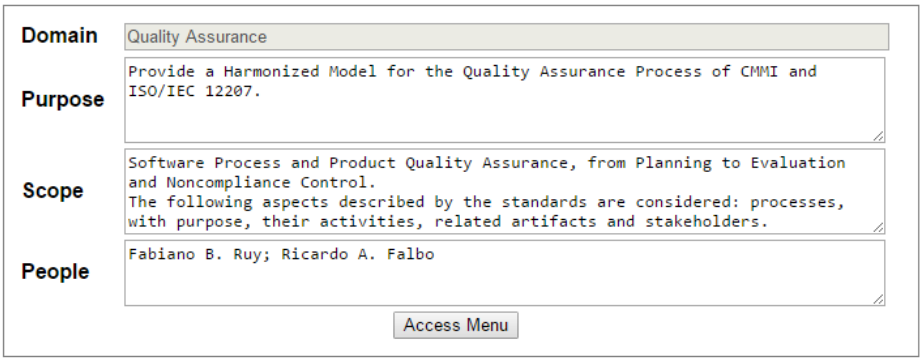


Figure 5.10. Mapper: Initiative Info.

(2) Astah Parsing

This action is responsible for reading the model information from the Astah file. Some instructions are given for organizing the input file to be parsed, as Figure 5.11 shows. The parsing process reads all the relevant models (structural, content and ontology models) and extracts their diagrams' images and their elements / concepts, definitions and relations. The diagrams are saved in a proper location and all relevant data is used for populating the tool model. This information is used for supporting the mappings in the next steps. In relation to the Harmony process, the parsing captures the results of the activities A1.4 (Select the SEON View), A2.1 (Develop the SSMs) and A3.1 (Develop the SCMs).

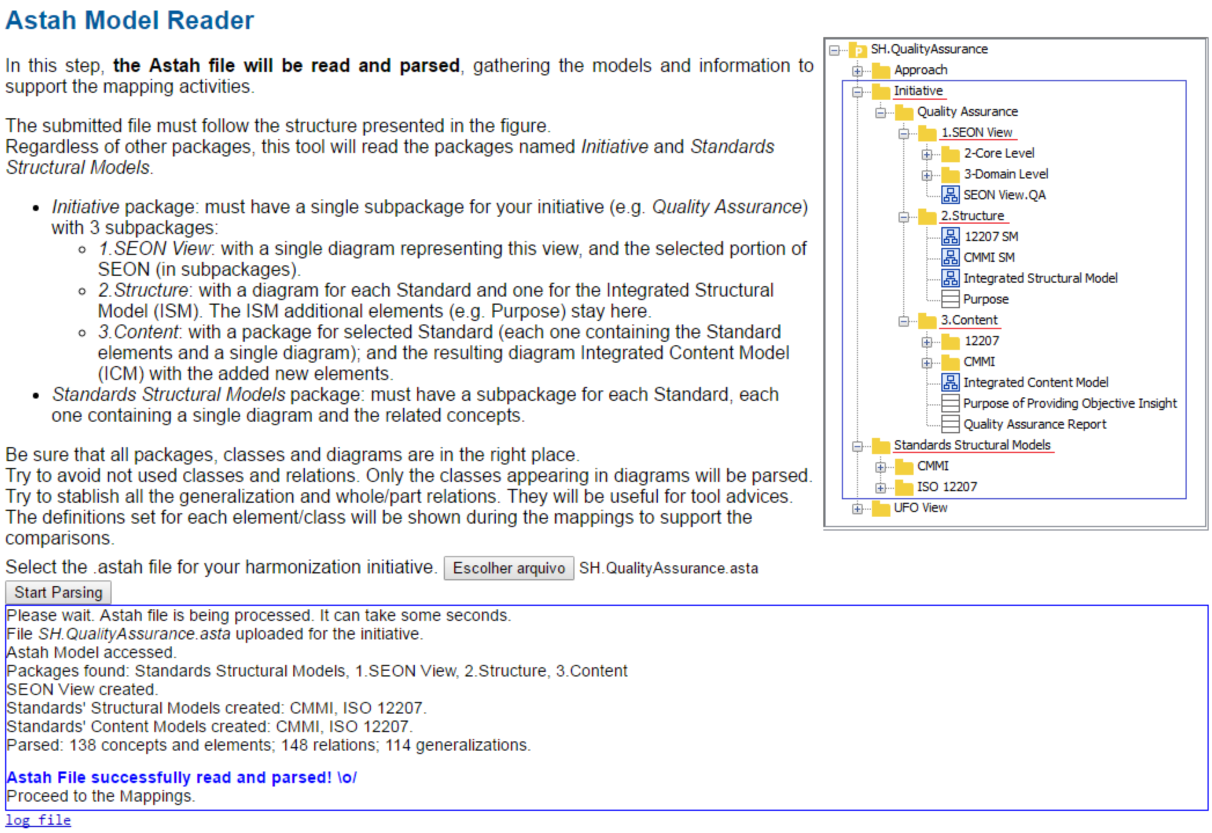
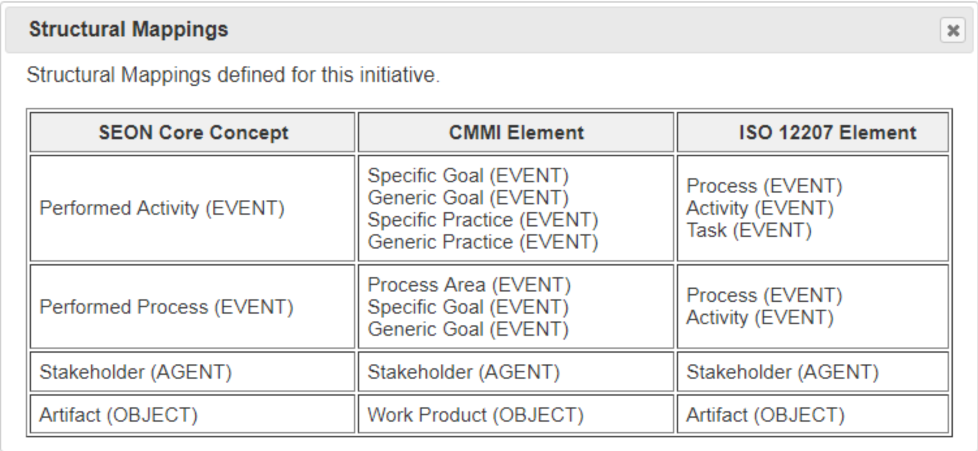


Figure 5.11. Mapper: Model Parser.

Once the Astah file is properly parsed, all the needed information is gathered and the user can proceed with the harmonization initiative.

(3) Structural Mappings

Mapper reads the structural mappings information from a text file and creates all the structural mappings and the corresponding matches. As explained, these matches are used to delimit the allowed content matches. Figure 5.12 shows the mappings informed to the tool in an initiative involving the CMMI-DEV and ISO/IEC 12207 standards. This action supports the Harmony activities A2.2 (Map SSMs to Core Ontologies) and A2.3 (Develop the ISM).



SEON Core Concept	CMMI Element	ISO 12207 Element
Performed Activity (EVENT)	Specific Goal (EVENT) Generic Goal (EVENT) Specific Practice (EVENT) Generic Practice (EVENT)	Process (EVENT) Activity (EVENT) Task (EVENT)
Performed Process (EVENT)	Process Area (EVENT) Specific Goal (EVENT) Generic Goal (EVENT)	Process (EVENT) Activity (EVENT)
Stakeholder (AGENT)	Stakeholder (AGENT)	Stakeholder (AGENT)
Artifact (OBJECT)	Work Product (OBJECT)	Artifact (OBJECT)

Figure 5.12. Mapper: Structural Mappings.

(4) Vertical Mappings

Vertical mappings (A3.2 - Map SCMs to Domain Ontologies) are fully supported by the tool. For each content model (SCM), Mapper allows creating a vertical mapping from the Standard’s Model elements to the SEON View concepts, guiding the establishment of the matches. Figure 5.13 shows, partially, the content mapping between the elements of the Quality Assurance process model from CMMI-DEV (PPQA) and concepts of the SEON View (QAO). This is the main screen of the tool. It was designed for easy visualization and selection of the notions, allowing more contextualized match decisions. The idea is to provide an abstraction so the user can work focused on the models using a simple interface. The user can see the source and target models and, for each selected notion, its type, relations, definitions and status. For creating a match, first the user selects the standard’s element (in the left-hand model) and the corresponding ontology concept (in the right-hand model). The associated information is provided in the bottom boxes. Then, the user selects the proper Match Type, the Coverage and provides related comments. Finally, the MATCH button establishes the match if all conditions are satisfied (e.g. element and concept specialize compatible structural notions, and the element is not already fully covered).

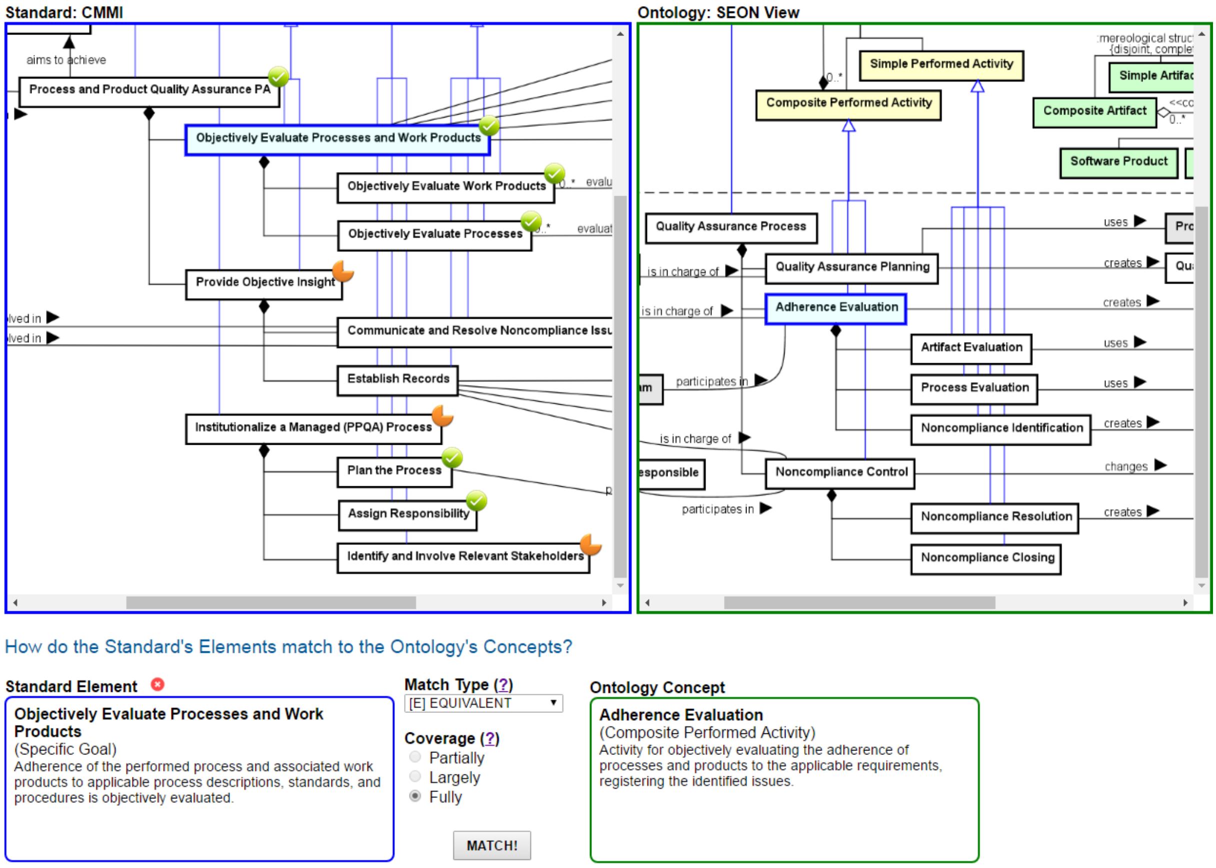


Figure 5.13. Mapper: Vertical Mapping.

As the matches are created, the standard's model view is updated, showing which elements are fully covered (), partially covered () or discarded from the initiative (). Additionally, the established matches are shown in a list, as Figure 5.14 illustrates, from which they can be removed (⊗) or have comments ({*C*}) edited. The coverage range is calculated based on the individual coverage of each standard's element. Finally, when all possible matches are done, a mapping analysis can be added by the user, to be included in the harmonization report.

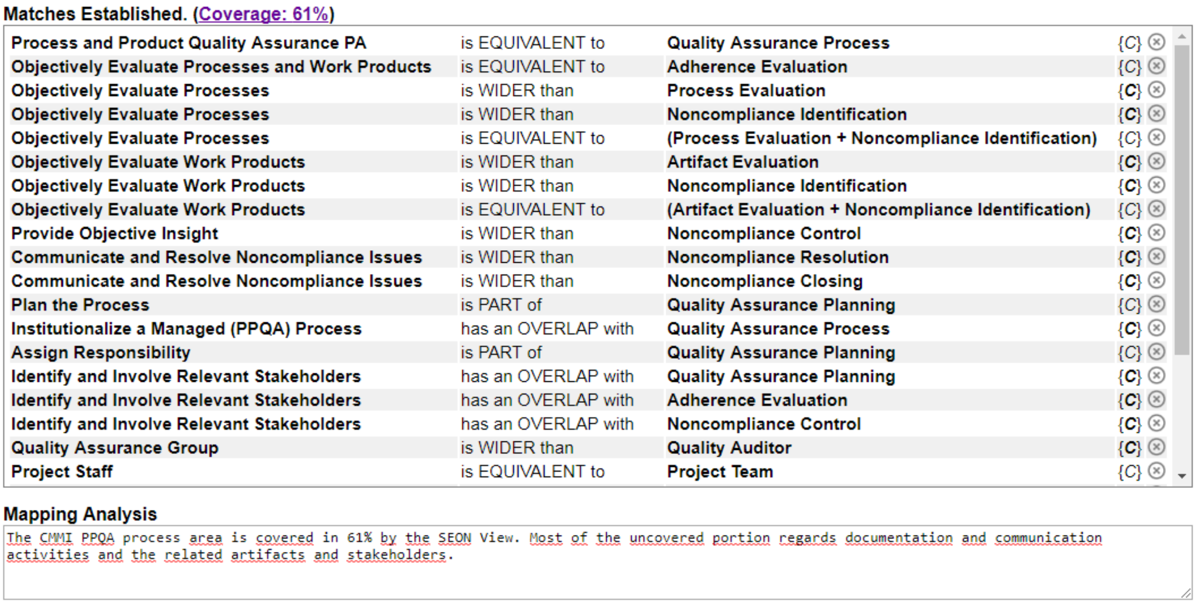


Figure 5.14. Mapper: Matches Established.

Figure 5.15 shows another view of the elements' coverage status, where the user can see the uncovered (-), partially covered ( ) and fully covered ( ) elements in a list (ordered by status and type). For the partially covered elements with multiple matches, the tool suggests Composite Matches, where the set of target concepts together can fully cover the source element, increasing the coverage.

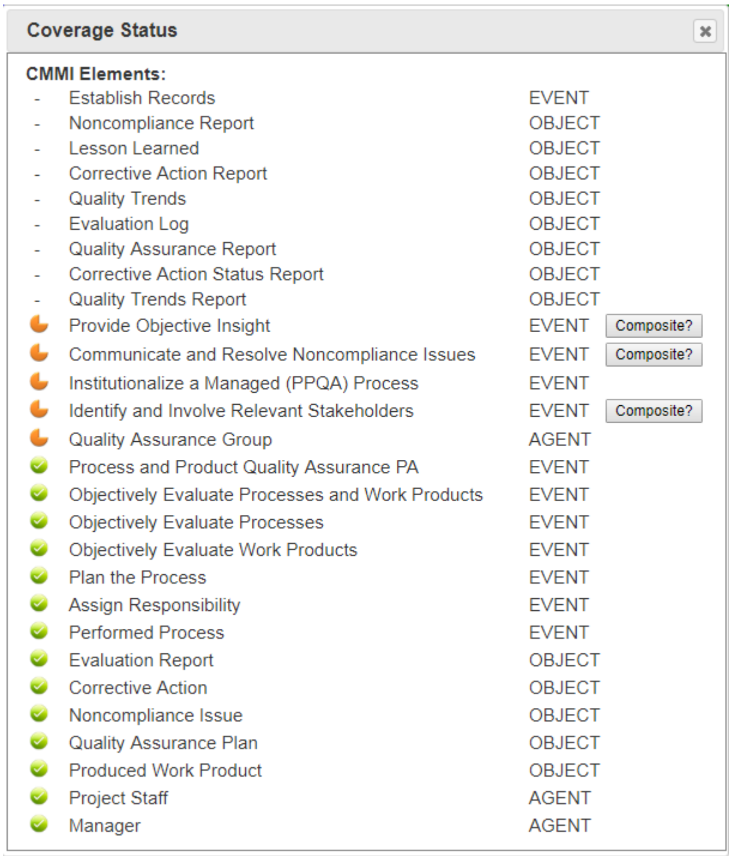


Figure 5.15. Mapper: Coverage Status.

(5) Horizontal Mappings

Once all SCMs are mapped to the SEON View, it is expected that all possible matches between the standards' elements and the ontologies' concepts are already established. However, diverse elements from the standards can remain partially or not covered (the SEON View may not cover some parts and specificities of the standards). At this point, the tool presents these uncovered elements and allows matching them to new elements created in the Integrated Content Model (A3.3 – Develop the ICM), i.e. establishing horizontal mappings. As Figure 5.16 illustrates, Mapper shows the lists of uncovered elements in each standard and the fields for creating a new ICM element. The user shall provide a name, a base type, a description, and all the matches with the related SCM elements (horizontal mappings).

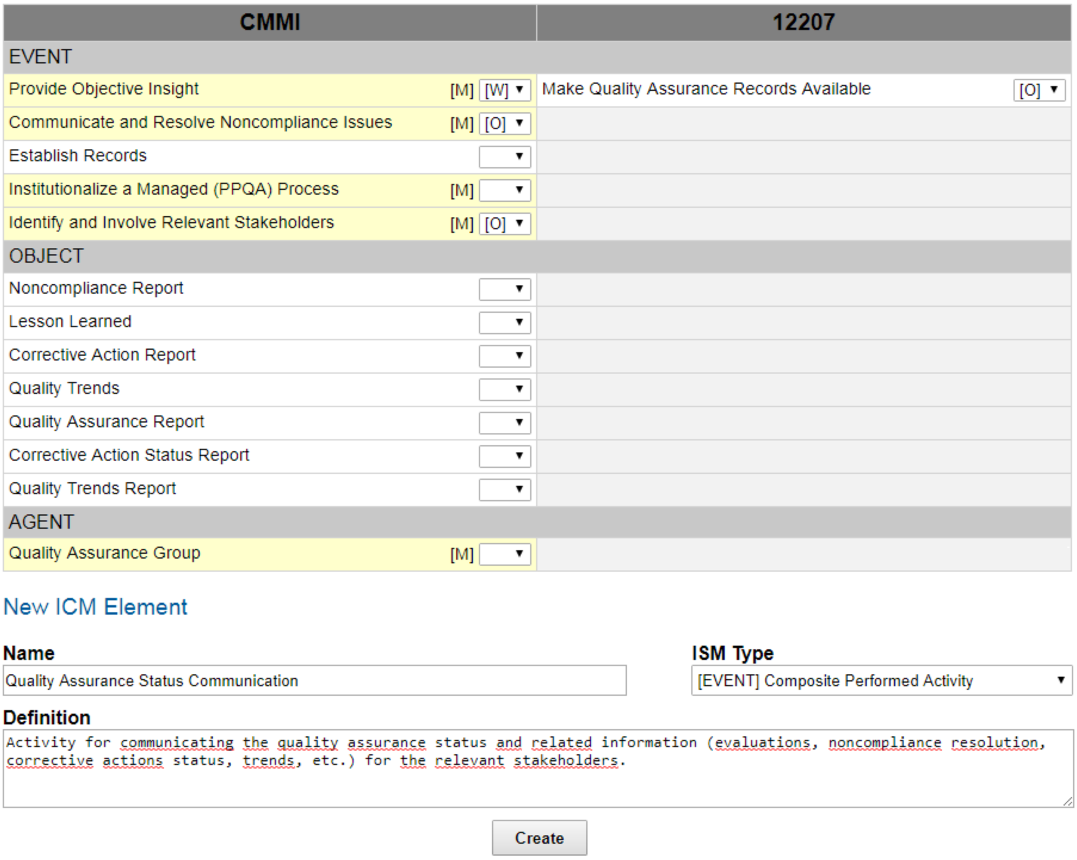


Figure 5.16. Mapper: ICM Mapping

In the presented lists, the background color of the rows represents the current coverage of the elements. The blank rows contain elements with no matches. The yellow rows contain the partially covered ones (current matches can be seen in the [M] symbol). Fully covered elements are not shown because they do not need new matches. Figure 5.16 illustrates the creation of a new ICM element, the *Quality Assurance Status Communication*. It is created by entering an ISM Type, a definition, and the matches with the elements selected in the list. The selection also informs the match type for each source element (one Wider, meaning that *Provide Objective Insight* in CCMI is a notion wider than the element being created, and three Overlaps meaning that the three elements selected are notions that overlaps the element being created). The result is presented as the last element in Figure 5.17, which shows the resulting new ICM elements with their matches (horizontal mappings). We can also see the coverage increasing with the new matches (e.g. in the Quality Assurance harmonization initiative, from 61% to 95% for CMMI, and from 90% to 92% for ISO/IEC 12207).

ICM Elements Created. Coverages: (CMMI: 61% + 34%) (12207: 90% + 2%)

CMMI: Noncompliance Report	[P]	**Quality Assurance Control Report** (Document)	Document describing the quality assurance findings and results for monitoring and controlling purposes.
CMMI: Corrective Action Report	[P]		
CMMI: Evaluation Log	[P]		
CMMI: Quality Assurance Report	[P]		
CMMI: Corrective Action Status Report	[P]		
CMMI: Lesson Learned	[P]	**Quality Assurance Trends Report** (Document)	Document describing the main lessons and trends of the quality assurance activities.
CMMI: Quality Trends	[P]		
CMMI: Quality Trends Report	[P]		
CMMI: Quality Assurance Group	[E]	**Quality Assurance Group** (Stakeholder)	Team responsible for supporting the objective evaluation and noncompliance control activities, by documenting and reporting the quality assurance findings, results and trends.
CMMI: Provide Objective Insight	[W]	**Quality Results Documentation** (Composite Performed Activity)	Activity for documenting quality assurance information such as the achieved results, lessons learned, quality trends, and noncompliances, producing the Quality Assurance Control and Trends Reports.
CMMI: Establish Records	[E]		
12207: Make Quality Assurance Records Available	[W]		
CMMI: Provide Objective Insight	[W]	**Quality Assurance Status Communication** (Composite Performed Activity)	Activity for communicating the quality assurance status and related information (evaluations, noncompliance resolution, corrective actions status, trends etc.) for the relevant stakeholders.
CMMI: Communicate and Resolve Noncompliance Issues	[O]		
CMMI: Identify and Involve Relevant Stakeholders	[O]		
12207: Make Quality Assurance Records Available	[O]		

Coverage Analysis | SAVE and Return to Menu

Figure 5.17. Mapper: New ICM Elements and their Matches.

It is important to highlight that Mapper only handles the models' information but it is not designed to create or update diagrams. Thus, the ICM Mapping shall be done iteratively, creating the elements in Mapper and modeling them in the modeling tool (Astah), where additional information (such as relations) can be included and the whole view of the ICM can be analyzed.

Finally, some elements can still remain not covered after the ICM Mapping. It happens due to several reasons, for example, because a standard's element can only be partially in the scope of the harmonization initiative, or all its relevant parts are covered but the element *per se* has not enough matches for fully covering it. In these cases, the tool provides an additional feature for analyzing these elements coverage and justifying these exceptions. At this point it is expected that the standard's elements are 100% covered by the ICM (or justified).

(6) Standard-to-Standard Mappings

As discussed in Section 5.4.4, the ICM and related matches can be used for helping to establish direct mappings between pairs of standards. Mapper supports this task in a very similar way it does for the

(4) Vertical Mappings. The main differences are: (i) the target model is another Standard's Model instead of the SEON View; (ii) there are variations in consistency verifications; and (iii) matches are suggested by means of inferences.

Since source and target standards' models were previously mapped to the same model (ICM), it is possible to reuse this information for deducing a set of matches based on the rules presented in Table 5.2. This mapping starts by deducing all possible matches for the involved pair of standards. Figure 5.18 shows the results related to the matches deduced in the Quality Assurance harmonization initiative, and some advices for proceeding with the mapping. It is important to check the deduced matches before proceeding, observing if they are consistent and if they express true matches. At this point, incorrect matches can point out errors in previous mappings or inconsistences in the standards.

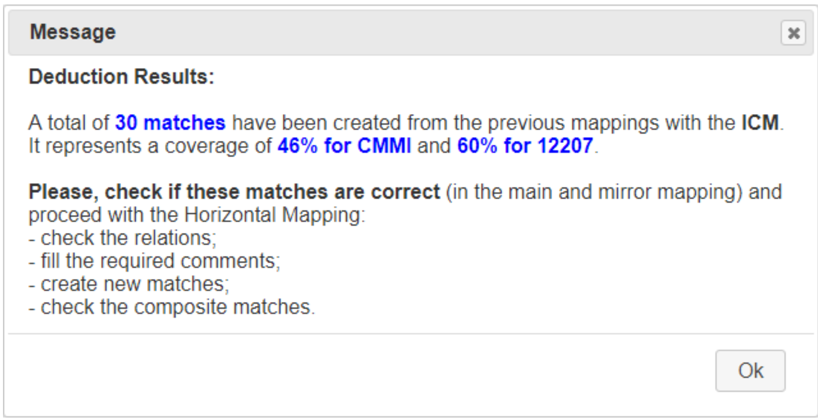
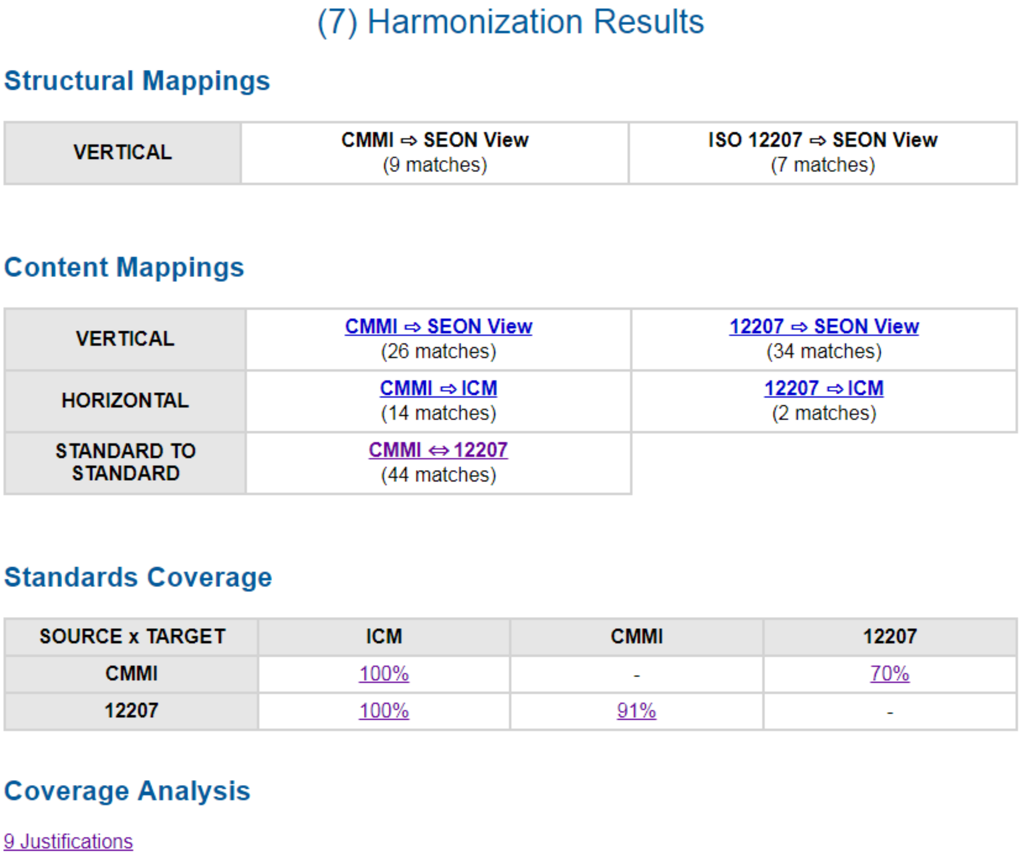


Figure 5.18. Mapper: Matches Deductions.

Once the deductions are analyzed, the user can proceed, completing the mapping by creating new matches. Since the matches are established in one direction (e.g. from Standard A to Standard B), for each established match, the tool creates another match in the inverse way. For example, if element A is matched as Wider than element B (A [W] B), then a complementary match is also created (B [P] A). In this way, it is possible to analyze the mappings in both perspectives, from Standard A to Standard B and vice-versa.

(7) Harmonization Results

Lastly, harmonization results are provided. Mapper builds a report containing all the mappings produced and related information (e.g. coverage, analysis and justifications). Figure 5.19 presents the beginning of this report with the summarized information and links to the detailed mapping tables with the established matches.



(7) Harmonization Results

Structural Mappings

VERTICAL	CMMI ⇨ SEON View (9 matches)	ISO 12207 ⇨ SEON View (7 matches)

Content Mappings

VERTICAL	CMMI ⇨ SEON View (26 matches)	12207 ⇨ SEON View (34 matches)
HORIZONTAL	CMMI ⇨ ICM (14 matches)	12207 ⇨ ICM (2 matches)
STANDARD TO STANDARD	CMMI ⇔ 12207 (44 matches)	

Standards Coverage

SOURCE x TARGET	ICM	CMMI	12207
CMMI	100%	-	70%
12207	100%	91%	-

Coverage Analysis

9 Justifications

Figure 5.19. Mapper: Harmonization Results

Observe that in the Quality Assurance harmonization initiative, the ICM covers 100% of both standards' models, and the CMMI model covers 91% of the 12207 model, while the 12207 model covers 70% of the CMMI model. The deductions represented 46% (out of 70%) of the CMMI model coverage, and 60% (out of 91%) of the 12207 model coverage, i.e. around 2/3 of the models coverage was deduced by the tool in this case, representing a considerable reduction in the mapping efforts.

5.6 Related Work

Harmony is an approach for harmonizing SE standards using ontologies and conceptual models to deal with semantic aspects. Like our approach, several works have dealt with standards harmonization along the last years, as discussed in Chapter 3 (e.g. (FERCHICHI et al., 2008) (JENERS et al., 2013) (PARDO et al., 2013) (HENDERSON-SELLERS et al., 2014)). These harmonization efforts focus on solving the same interoperability problem, sharing similar purposes and techniques. However, each one addresses the issue with a particular solution. This section discusses some of the key aspects present in harmonization approaches putting Harmony in comparison with the main related works.

Table 5.3 summarizes a comparison between Harmony and four of the most mature harmonization approaches related to our work. The approaches are compared taking into account six main aspects:

- **Defined Process**: does the approach define a process for conducting harmonization initiatives?
- **Resulting Products**: what are the main products of a harmonization initiative?
- **Techniques Applied**: which are the adopted techniques and how are they applied?
- **Knowledge Support**: how does the approach use a knowledge support?
- **Semantic Treatment**: what are the actions taken for dealing with the standards' semantics?
- **Supporting Tool**: does the approach provide software tools for supporting the initiatives?

Table 5.3. Harmonization Approaches Comparison.

ASPECT	Ferchichi et al.	Mosaic (Jeners et al.)	HFramework (Pardo et al.)	ISO Initiative (Henderson-Sellers et al.)	Harmony
Defined Process	General Process with main steps	No Process, but detailed technical instructions	Process, and methods for managing and performing harmonization	Guidelines for model derivation	Process with detailed guidelines
Techniques Applied	Homogenization, Mapping, Integration	Homogenization, Mapping, Integration	Homogenization, Mapping, Integration	Homogenization, Integration	Homogenization, Mapping, Integration
Resulting Product	Integrated Model	Integrated Model	Integrated Standard	Derived Ontologies	Integrated Model
Knowledge Support	Standards' Structure Ontology	Standards and Integration Metamodels	Harmonization and Process Ontologies	Ontological Infrastructure	Ontology Network
Semantic Treatment	Object Model based on an Ontology	Metamodels, Elements Classification	Ontologies, Mapping and Integration Criteria	Ontological Infrastructure Derivation	Foundational Grounding, Ontology Mappings, Match Types
Supporting Tool	None	Concepts Extraction, Similarity and Comparison Tools	HProcessTOOL, for Harmonization Management	None	Mapper, a Mapping and Integration Tool

Presenting a well-defined **Process** is important for a harmonization approach to enable its proper execution by those interested. All the approaches in Table 5.3 present at least some instructions for conducting the harmonization efforts. Ferchichi and colleagues (2008) provide a general four-steps process giving the main guidelines for harmonizing multiple standards. The Mosaic approach (JENERS et al., 2013) does not present a step-by-step process, however they further discuss, providing relevant technical details, how the harmonization is conducted using a set of models and metamodels. HProcess, from HFramework (PARDO et al., 2013), is the most complete process we have analyzed. It is a process for managing and performing harmonization initiatives, from the organization's goals to the harmonized results, including templates and methods for homogenization, mapping and integration

(HMethods). The ISO harmonization initiative (HENDERSON-SELLERS et al., 2014) does not establish a process, but provide instructions for building the proposed ontological infrastructure by means of ontology derivation. Lastly, Harmony's Process is composed of three main phases with activities providing guidelines (including technical, practical and semantic related details) on how to conduct a harmonization initiative. One of its advantages is offering detailed instructions and tips for being performed by other people, including users with different profiles and purposes (such as standards' developers and organizational quality teams). Like HFramework, Harmony is supported by a software tool to aid its main activities.

The approaches are very similar regarding **Techniques** applied. The triad homogenization-mapping-integration is frequently adopted. The exception is the ISO Initiative, which, from a higher-level ontology, instantiates ontologies (homogenization) representing a set of standards (integration), not explicitly considering the mapping technique. The other approaches use the three techniques, however with some peculiarities. In general, Homogenization adapts the standards' contents to a predefined format to facilitate comparisons; Mapping consists in lower-level comparisons between the standards' elements, establishing links (matches); Integration defines a new (integrated) schema combining the selected portions of the involved standards.

Ferchichi et al. homogenize the standards by modeling them as instances (objects) of a predefined metamodel. This object model also includes the mappings between related practices (including a correlation level: *weak* / *medium* / *strong*). Finally, the object model is used as basis for building an integrated model. Jeners et al. use a more elaborated support of models. Homogenization is achieved by representing each standard in a model derived from a standard's metamodel. An interesting point here is that the model elements are classified according to their types (e.g. *activity*, *artifact*, *role*, *context*) for supporting comparisons. Mapping and Integration occur simultaneously, by relating and merging the standards' model elements into an integration model (also derived from a metamodel). Pardo et al. perform homogenization using a template (in a tabular format), built from a process-reference ontology, to be filled with the standards' contents. Unlike the Mosaic approach, HFramework is more focused in the standards' practices (textually described) than in their concepts (such as *activities*, *artifacts* and *roles*). Mappings are conducted for each pair of the involved standards, comparing all the practices and setting a relationship degree (*not*, *weakly*, *partially*, *largely* or *strongly related*, with percentage values). Integration is supported by the mappings, textually integrating the practices descriptions for building an integrated standard. Several instructions and criteria are given to support applying the techniques.

Harmony also applies the three techniques with some similarities and differences with the other approaches. Similar to the Mosaic proposal, we also use a base model for building the standards' content models. However, instead of a single metamodel, Harmony presupposes the structural representation of each standard (SSMs), from where the content models are derived. Hence, the

Homogenization occurs indirectly: when the structural models are mapped to a common ontology (SEON View), each structural element is “classified as” (mapped to) a core concept. Thus, the specialized content elements carry this classification, which is used forward for mapping and integrating them. Although this solution does not provide a homogeneous representation for all standards in the beginning of the approach, it has the advantage of keeping a representation most faithful to the original standards, avoiding semantic losses in transformations. Moreover, the structural and content standards’ models are relevant artifacts with a high potential for being reused in other harmonization initiatives. For example, the CMMI-DEV and ISO/IEC 12207 structural models (and their mappings) were reused in several applications of Harmony. Mappings are also heavily based on models. Each content model is mapped to a common ontological model. The notions comprised in the standards’ and ontological models (already classified according to their structures – structural elements or core concepts – and natures – UFO ontological types) are compared and matched, including the nature of the relation (from eight match types) and a coverage extension (*partially*, *largely*, *fully*). A number of checks is made for assuring consistency. Finally, Integration is performed by producing an integrated model based on the ontological model and previous mappings. The result is the ICM, an integrated view comprising all the relevant portions of the involved standards considered in the initiative scope.

The **Results** of any harmonization approach should provide a unified view considering the relevant aspects of the involved standards. As said, all the related approaches produce such view in an integrated model with different formats. Ferchichi et al. create that model from an object model instantiated for representing and mapping the standards. Although they do not specify a format for the final result, a process cartography unifying elements of the standards is shown. Jeners et al. produce an Integration Concept Model merging the elements of the involved standards and keeping the traceability to them. Pardo et al. build an integrated model as a textual standard-like artifact, unifying the standards’ practices. Henderson-Sellers et al. proposal considers the addition of ontologies to their infrastructure representing a standard or a group of related standards. Harmony’s main products are the ISM (Integrated Structural Model) for structural harmonization and ICM (Integrated Content Model) for content harmonization. They are conceptual models derived from ontologies and completed with combined elements for representing the portions of the involved standards not addressed by the ontologies. ICM is similar to Jeners et al.’s Integration Concept Model, since both are conceptual models integrating the standards’ contents and keeping the mappings to them. While our ICM came from an ontology (with domain knowledge and grounded in higher level ontologies), the Jeners et al.’s one is based on a simpler metamodel defining the admissible types and relations. Another advantage in our solution is that the ICM relations (matches) with the standards’ elements are semantically richer as they carry information on their types and coverage, and comments. Although an integrated standard (like the one produced by HFramework) could be a more natural result of an approach integrating standards (and more easily used in some situations), the integrated content model offers other benefits. It provides a holistic view of the solution that facilitates its analysis and exploration in many contexts. Moreover, models are easier to check, manipulate and reuse. They can even be transformed in other artifacts, such as a process or an integrated standard (with some effort, of course).

Dealing with standards harmonization is a knowledge intensive activity. Besides the user experience, the approaches should count on some kind of **Knowledge Support** for being successful. All the analyzed approaches are aware of this and, in different ways, use some knowledge source. Ferchichi et al. is one of the first attempts of using ontologies as a common model for standards harmonization. They apply a simple model representing the knowledge of the standards' structure to instantiate their standard's object models. Henderson-Sellers et al. proposal is, in most part, an ontological infrastructure containing a set of ontologies in multiple levels for representing the ISO standards in different abstraction levels. This infrastructure is expected to be used as basis for reviewing and creating new ISO standards. Pardo et al. use two ontologies for defining the harmonization and process-reference knowledge in their approach. Although the ontologies are not directly applied during their initiatives (e.g. for deriving other models), they were used for building the approach itself, being the basis for defining templates and methods. Jeners et al. use two metamodels for instantiating the standards' models and the integrated model. These metamodels include different types for classifying the elements according their nature, which is useful for the harmonization activities. Finally, Harmony was built considering SEON, an ontology network with domain, core and foundational concepts providing a comprehensive knowledge support in different contexts, as follows. The domain ontologies are used as a common model, bearing the domain reference knowledge, in the content mappings and integration. The standards' content models are directly mapped to them. Core ontologies support the structural harmonization and provide the core concepts for classifying the standards' elements. A foundational ontology, besides grounding the lower-level ontologies, gives the basic ontological types (e.g. *event*, *object*, *agent*) for supporting the structural and content mappings and integration. As pointed out in the empirical studies, most of the harmonization efforts are done counting on the ontologies' consistency and semantics. Moreover, Harmony requires the user to build (or reuse) the standards' structural models, adding a new knowledge support on the specific standards for the onward activities.

Since standards harmonization is a semantic interoperability problem, dealing with diverse sources of processes and practices, often not sharing a common conceptualization, it is the greatest challenge in this topic. Thus, the approaches must consider a **Semantic Treatment** for the information being handled. The standards should be understood and the meaning of their elements took as the main aspect for analyzing, mapping and integrating them. The previous discussion mentions diverse situations where semantic aspects are addressed by the approaches. Ferchichi et al. use an ontology for establishing a basic organization for modeling the standards' elements. A correlation level is provided by the user during the mappings. Henderson-Sellers et al. propose an entire ontological infrastructure describing the standards' information in different abstraction levels. Each new ontology is built based on a (set of) standard(s) and derived from a higher-level ontology. Pardo et al. apply the knowledge about the process and harmonization domains (represented as ontologies) for defining their methods. Standards are homogenized using a template for classifying the information, and several instructions are given for the user to deal with the target information meaning (e.g. instructions and criteria on how to write an integrated practice from two related practices from distinct standards, in varied contexts (PARDO, 2012)). Jeners et al. model the standards' contents using a common metamodel, which classifies the standards' elements and the relations between them. The integrated model is also built according to a metamodel including some relation types. The types of standards' elements are used during the mappings and integration for relating only the compatible elements. They also explore other techniques making use of algorithms for concept extraction, similarity metrics, and comparisons (JENERS; LICHTER, 2013). In Harmony, semantics is one of the key aspects. As explained, SEON provides the needed knowledge support for dealing with the content and structural standards' elements. Moreover, semantic aspects are also methodologically addressed. Harmony process requires the use of ontologies as a semantic referential. A structural harmonization phase prepares a set of models and connections for enriching the content harmonization activities with the semantics provided by the ontologies. The activities are conducted analyzing the elements' nature, type, relations and meaning, encouraging a semantic judgment. Instructions are given for modeling, analyzing, comparing and integrating the standards' elements, considering their meanings and contexts. Different match types are provided for dealing with distinct types of relations and notion natures, establishing semantically richer connections. The resulting integrated model itself is built from an ontology and has all elements linked to the source standards' elements. Furthermore, the use of a foundational ontology provides a well-founded ground to the ontologies, helps to classify also the standards' elements according to basic distinctions, and supports the selection of the proper match types. All the notions and matches are directly or indirectly related to UFO foundations. It is an important differential in a context where distinct standards (or even the same standard) assign different meanings to the same element (PARDO et al., 2012) (RUY et al., 2014) (HENDERSON-SELLERS et al., 2014) (RUY et al., 2017).

Specialized **Supporting Tools** provide relevant assistance in standards harmonization activities. The most important efforts, characterized by semantically oriented decisions are still dependent of human judgment. However, many supporting tasks can be automated for reducing the comparison, mapping, integration and management efforts by making consistency checks, processing related data and providing the expert with relevant information and views to take the right decisions. Although the five analyzed approaches recognize the importance of automated support, only three of them have developed some kind of tool. Pardo et al. propose HProcessTOOL (PARDO et al., 2001), a web tool focused on the management of the harmonization initiatives. It is based on their harmonization process and helps to identify, define and configure the strategies that are suitable for putting multiple models into consensus and harmony. Jeners et al. explore more technical aspects of harmonization, proposing automated concept extraction (JENERS et al., 2012), identification of similarities using metrics (JENERS et al., 2012b), and automated comparisons (JENERS; LICHTER, 2013). The tool is able to automatically compare selected procedures, obtaining a similarity degree, and then more information on dependencies and coverage and adoption degrees. The Harmony's tool, Mapper, supports the content mapping and integration activities of the proposed approach. Like the harmonization process, it is based on the standards' and ontological models, providing the information in a diagrammatic view. Mapper imports the previously created models from a general modeling tool (Astah) and visually presents the pairs of subject models and related information for the expert, so she/he can take the proper harmonization decisions concerning mapping or integration. In this process, it performs consistency checks and, finally presents the coverage results in a report. Each of the three discussed tools provide, even distinct, clear advantages to their respective approach, speeding up the harmonization efforts and making the results more consistent. While HProcessTOOL works in the management of the initiative, Mapper deals with semantic matches for mapping and integration activities, and the Jeners et al.'s tool focus on more detailed tasks such as concepts extraction, similarities and comparisons.

In sum, standards harmonization demands a semantically-oriented solution, counting on a robust knowledge framework and a tool supported approach encouraging an effective semantic analysis. We believe this is the main Harmony's differential in relation to the other analyzed approaches, consisting in a step forward in the standards harmonization topic. However, our approach is new and still has some limitations, discussed in the final chapter.

5.7 Final Considerations

This chapter presented our ontology-based approach for harmonizing SE standards. Harmony has been developed for addressing a semantic interoperability problem: the harmonization of multiple standards. The resulting approach is founded on a literature mapping, experiences in SPI projects, analysis of preliminary results, diverse discussions, and a number of empirical studies.

The main focus of Harmony is dealing with the semantics of the standards. The meaning of the information described by the standards should be put in evidence for analyzing and harmonizing them. For reaching this goal, many aspects were addressed regarding the models used for representing the standards' structures and contents, referential domain knowledge coming from the SEON ontologies, foundational ground applied for the notions handled, harmonization techniques applied, distinct types of matches and their properties, and the resulting integrated model. Moreover, Mapper supports Harmony application, automating mechanical actions, allowing the user to focus on the most relevant aspects, those requiring a semantic analysis, such as examining the standards' elements for establishing the proper matches.

The approach was evaluated along the last semesters, being applied in harmonization initiatives in three empirical studies. These studies provided a valuable feedback for identifying problems and improvement opportunities in Harmony. The empirical evaluations are discussed in the next chapter.

Finally, we believe Harmony fills a research gap regarding semantic aspects not addressed by other approaches. However, it is only a step ahead. Standards harmonization is still a challenging topic demanding research and the effective adoption by standardization and software organizations.

6 Evaluating Harmony

This chapter addresses the empirical studies we have conducted for evaluating Harmony. It presents three evaluations performed along the last semesters of this research, discussing their results and main findings used for evolving the approach.

6.1 Introduction

Harmony was applied in different moments and situations throughout its incremental development. Its use and related feedback have been essential for its evolution. Although we have not yet applied Harmony in a real industry project, it was experienced in some contexts, varying the subject domains, involved standards, participants' profile, and harmonization purposes. In this sense, our main efforts focused on producing the proof-of-concept and conducting three empirical evaluations: (i) with a preliminary version of Harmony for producing harmonized processes from SE standards; (ii) with the first version, resulting in mapped and harmonized standards' models; and the last one, (iii) with the current version, also creating harmonized standards' models, but involving an expert (since the other two were conducted with graduate students).

The empirical studies for evaluating Harmony were designed and conducted following the guidelines, presented in (OATES, 2011). In short, they consisted in:

(i) Presenting Harmony (and SEON) for the participants;
(ii) Defining the main parameters for each initiative (purposes, specific standards and processes to be harmonized, expected results);
(iii) Performing the harmonization initiatives;
(iv) Evaluating the results produced;
(v) Applying questionnaires and interviews for collecting feedback from the participants.

During the studies some support was provided to the participants for elucidating the use of Harmony and the tool. After each experiment, the results obtained and participants' feedback were discussed to identify the main limitations and improvements. In general, the application of the approach by other people in the studies provided us with relevant information for evolving Harmony, Mapper, and even SEON.

Chapter 5 presented Harmony and results produced during its first application in a proof-of-concept harmonizing Quality Assurance processes. This chapter describes Harmony applications by other people in three empirical studies. Section 6.2 describes the Studies 1 and 2, experiments with graduate students using previous versions of Harmony. Section 6.3 discusses the Study 3, performed by an expert using the current Harmony version. Section 6.4 presents the final considerations.

6.2 Experiments with Previous Versions of Harmony

Before achieving the Harmony version presented in this text, two experiments were conducted using previous versions of the approach. Both experiments were performed in the context of graduate courses as part of the students' activities. It indicates that the participants had time and motivation enough for developing the proposed initiatives. These experiments were applied mostly for evaluating our methods (still in definition) and also for exploring some directions and ideas on harmonization. The main characteristics and results of the evaluations are discussed in the sequel.

6.2.1 Empirical Study 1: An Experiment with Harmony's Pre-release Version

The initial experiment took place during the first semester of 2016, as part of the course "Software Engineering", an advanced course for graduate students. Harmony was being developed, in a pre-release stage, before the introduction of models for representing the standards as well the development of Mapper. The goal of the experiment was to explore the basis defined until that moment for Harmony. All the work, including the mapping efforts, was made from the standards' texts, organized in supporting tables. The purpose of the initiatives in this experiment was to harmonize, for each subdomain, the related standards producing a software process definition, adherent to all the involved standards. Three SE subdomains were considered (Requirements Development, Construction, and Testing), each one involving the respective processes of three standards: CMMI-DEV (SEI, 2010), ISO/IEC 12207 (ISO/IEC, 2008) and SWEBOK (BOURQUE; FAIRLEY, 2014).

Three groups were defined, one per domain, composed of three or four students. All participants had an IT degree and declared academic-only experience with the main involved topics (SE standards, ontologies and the specific SE subdomain). Each group conducted one initiative, where the students selected the proper standards' portions; extracted their information, organizing it in tables; and mapped it to the concepts of a selected SEON View. From these mappings, each group defined a software process (with activities, roles, inputs and outputs) considering the harmonized information.

Although the pre-release version of Harmony was simpler than the current one, the efforts for extracting, analyzing, mapping, and integrating the standards' contents were definitely higher than using the current version. The fact is that handling a high volume of text in diverse activities is not feasible for our purposes. Besides the participants' efforts producing the planned results, we also had a substantial work for checking and evaluating the resulting mappings and processes.

From this experiment results, participants' feedback and subsequent discussions, we have learned four main lessons:

(i) Ontologies indeed provide a valuable support for the harmonization efforts;

(ii) The standards' information must be better organized and represented (e.g., using models);

(iii) The standards' contents would be easily handled with the support of the standards' structures; and

(iv) Some kind of automated tool is essential for making the work easier, as well as for checking consistency.

6.2.2 Empirical Study 2: An Experiment with Harmony's First Version

From the lessons learned in the Empirical Study 1, we evolved Harmony, creating its first version, similar to the one presented in this thesis. That version was applied in a new experiment, the Empirical Study 2. The main differences from the Study 1 regard: (i) the intensive application of conceptual models; (ii) the use of Mapper for supporting the mapping activities; and (iii) the harmonization initiative's results, producing now mapped models and an integrated content model (ICM).

This experiment was conducted during the second semester of 2016, as part of the course "Ontologies for Software Engineering", an advanced course for graduate students. It involved eight students organized in three groups according to the process in focus: Configuration Management, Requirements Development, and Software Design. The standards used as basis were: CMMI-DEV, ISO/IEC 12207, and ISO/IEC 29110 (ISO/IEC, 2011). All the three initiatives had the purpose of providing a harmonized model for the respective process taking these standards into account.

The participants, five master degree students and three doctorate students, all with computing degrees, declared an average low experience level (mostly academic) on the main aspects involved in the activity development: SE Standards, Conceptual Models and Ontologies, and the target SE Subdomain. Although the low experience level could affect the results, it was a good opportunity for evaluating the approach's guidance and clarity. It is worth to mention that one of these students also participated in Study 1, being a good source of information for comparing the Harmony versions and advancements. Moreover, the related difficulties were mitigated by providing a harmonization example (the Quality Assurance initiative presented before) and some supporting meetings. As result, two of the groups produced good quality results (mappings and integrated models).

Besides analyzing the initiatives' results, we have applied individual questionnaires and group interviews, using the scripts provided in Appendix B. In short, the feedback collected includes the participants' profiles, information related to the difficulties for performing Harmony activities, and the contributions of different aspects of the approach.

When asked for the difficulties to perform the initiative ("*What was the difficulty degree on understanding and applying the Approach / Ontologies / Standards / Conceptual Models / Tool involved?*"), we collected the information summarized in Table 6.1 and described in the following.

Table 6.1. Reported Difficulties.

Aspect / Difficulty	Very Easy	Easy	Medium	Hard	Very Hard
Approach: understanding and application	-	62%	12%	12%	12%
Ontologies: understanding, selection and application	25%	25%	37%	12%	-
Standards: understanding and information extraction	-	-	25%	50%	25%
Conceptual Models: creation	-	25%	12%	37%	25%
Tool: understanding and application	25%	12%	50%	12%	-

- Understanding and applying the **Approach** received an average ***medium*** degree of difficulty. The main problems occurred in activities involving selecting, understanding and comparing standards' elements, due to the knowledge needed to identify the appropriated parts of the standards and to compare them. Another difficulty mentioned was related to the approach abstraction level, said overcome by the provided example.
- Understanding, selecting and applying **Ontologies** were considered ***easy*** tasks, with no relevant problems, and with SEON being pointed as a consolidated source of domain knowledge, which helped to understand diverse standards' aspects.
- Understanding and extracting information from the **Standards** were considered ***hard*** tasks. The main reported difficulties address the lack of experience in dealing with standards, and aspects related to the standards' descriptions: ambiguity, implicit information, and lack of uniformity.
- Creating **Conceptual Models** was also considered a ***hard*** task, mainly due to their close relation with the standards. The main reported difficulties were related to understanding and extracting information from the standards. The only issue directly related to the models was the effort to build and navigate large models.
- The **Tool** was considered ***easy*** to understand and use, pointed as intuitive and helpful. The main reported difficulties were related to the tool initial instability, the restrictions for reloading conceptual models (causing some mappings reset), and the navigation in large models.

Analyzing the reported degree of difficulty per approach step ("*What was the difficulty degree to perform each Approach step?*"), the activities considered hardest were those requiring a further understanding of the standards and the meaning of their elements (current activities A1.3 – Select the Standard's Portions, A2.1 – Develop the SSMs, and A3.3 – Develop the ICM). As expected, the low degree of experience of the participants regarding standards has affected some activities. An important exception is step A3.1 – Develop the SCMs, considered as medium difficulty, where the participants had to create conceptual models from the standards' contents; however, with the structural models' support, it became kind of an automated activity, with reliable results in general.

When asked for the contributions of different aspects for performing the activities ("*What was the contribution level of the Ontologies / Approach / Tool for the harmonization initiative?*"), most of the participants considered that the provided Ontologies, Approach and Tool had a ***high*** or ***very high*** level of contribution for the harmonization initiatives. The collected information is summarized in Table 6.2 and commented in the following.

Table 6.2. Contributions from the Different Aspects.

Aspect	Very Low	Low	Medium	High	Very High
SEON Ontologies	-	-	-	25%	75%
Harmonization Approach (Harmony)	-	-	12%	37%	50%
Mapping Tool (Mapper)	-	-	12%	12%	75%

The participants reported that the **SEON Ontologies** have contributed by: (i) providing a better understanding of the domain, and aiding understanding also the standards' portions; (ii) supporting the identification of elements during the standards' information extraction; (iii) guiding the mappings; (iv) offering clear and well defined concepts; (v) presenting concepts in a uniform granularity level; and (vi) allowing deductions of the relations (by humans) and matches (automatically and by humans).

Concerning **Harmony**, the main reported contributions were: (i) providing gradual steps for conducting the harmonization initiative, breaking a complex problem into feasible tasks; (ii) supporting the definition of the content models by specializing the structural models; (iii) performing the mappings and harmonization activities from conceptual models of the standards (much easier than dealing with text); (iv) providing a defined set of match types between the elements; and (v) systematic reusing knowledge from one step to another (e.g. scope for defining the models, structural models for defining the content models, ontologies for creating ISM and ICM). When questioned for suggestions, the participants asked for two new match types, allowing *specialization* and *generalization* matches, which was found useful when dealing with artifacts and agents. These match types were included in the current approach, together with the *act as* and *is acted by* match types.

Mapper has contributed by: (i) offering a sound support for the mapping activities, allowing the user to focus on the elements' meanings and matches; (ii) providing an easy-to-use and productive mapping interface where the elements could be analyzed and selected directly from the models' diagrams; (iii) doing a number of consistency checks; (iv) offering a view of the progress and coverage; (v) calculating the uncovered elements and deducing matches; and (vi) presenting the mappings and harmonization results in an easy way to interpret and identify previous mistakes. Some suggestions for the tool were: (a) allowing collaborative work; (b) allowing to save and export data; (c) preserving the unaffected matches after reloading new models; (d) entering the coverage percentage in a partially covering match; (e) improvements in the composite match suggestions; (f) enriching the tool decision points with more related information (definitions, previous matches, relations etc.); and (g) including features for discarding elements perceived as (partially or totally) out of scope. Items (b), (d), (e), (f) and (g) were partially or totally addressed in the current tool's version.

This experiment was important for identifying the approach aspects that were working satisfactorily and those requiring improvements. Considering the evaluation of the results produced, the participants' feedback, and a general analysis of the initiatives, the main conclusions are:

- The **ontologies** provided a valuable support by offering domain grounding and a well-structured knowledge reference for the mappings;
- Mapping standards' contents from **conceptual models** is a much easier, verifiable and less wearing activity than from text;
- The addition of **structural models** to the approach made the content modeling activities easier (it could be even semi-automatized in a future work);
- The **tool** was fundamental for reaching more detailed and consistent results.
- The participants' lack of experience with the involved **standards** increased the harmonization efforts and affected some of the results.

The identified problems were taken as improvement suggestions, mainly for the approach (but also for the tool) and for making the guidelines more complete and clearer, evolving Harmony. However, we considered that a more advanced experiment was needed. The approach should be evaluated by a **domain expert**, with a deeper experience mainly in the subject domain and in the standards being harmonized. This way, we could evaluate Harmony applicability when it is used by people with a profile closer to who develop and maintain quality standards.

6.3 An Experiment with Harmony's Current Version

The last empirical study took place during the first semester of 2017, involving a Software Testing domain expert. She has a Computer Science Ph.D. and is professor in the Software Engineering / Software Testing area, with more than 10 years of experience, including professional and research projects. The Software Testing initiative harmonized the standards ISO/IEC 12207, ISO/IEC 29110 and ISO/IEC 29119 (ISO/IEC, 2013c). It applied the versions of Harmony and Mapper described in this thesis (except for some improvements made in the text organization and descriptions, not implying in significant changes to the process). In the following, we present some information about the initiative and an analysis of the main results. The complete harmonization report is found in Annex A.

6.3.1 Empirical Study 3: Harmonizing Software Testing Processes

This evaluation aimed at harmonizing Software Testing Processes of three ISO standards, namely: ISO/IEC 12207, ISO/IEC 29110 and ISO/IEC 29119. ISO/IEC 12207 is a general software standard, providing a comprehensive set of life cycle processes, activities and tasks. Although it has a Software Qualification Testing process, other information regarding tests is presented in different parts of the standard, such as in Software Construction, Software Integration, Verification and Validation Processes. ISO/IEC 29110 is to be used by very small organizations for establishing processes to implement development approaches considering some activities. Its main testing tasks are established in the Software Architectural and Detailed Design, Software Construction, and Software Integration and Tests activities. Unlike the other two standards, ISO/IEC 29119 is specific for software testing. Its purpose is to define a set of standards for software testing that can be used by any organization. It is the most detailed standard, presenting several testing processes with organizational, management and dynamic testing activities (RUY et al., 2017b).

The Software Testing harmonization initiative had the purpose of *providing a harmonized model for Testing Processes* of the involved standards. The scope was defined as *Dynamic Testing Processes, excluding aspects related to Test Management (such as test planning and monitoring) and Test Environment (e.g., test environment set-up and maintenance), as well as test organizational aspects (such as organizational test polices and strategies)*. The portions of the three standards were selected according to this scope. The selected SEON View comprises mainly process-related concepts from ROoST (Reference Ontology on Software Testing) (SOUZA et al., 2017).

In the Structural Harmonization phase, the structural models (SSMs) for ISO/IEC 12207 and ISO/IEC 29110 (as well as some matches to core concepts) were reused from previous initiatives, and the SSM for ISO/IEC 29119 was built from scratch (by the expert with our review). In the following, the structural mappings were established by creating correspondences between the structural elements of each standard and the SEON View core concepts, as presented in Table 6.3. Finally, the integrated structural model (ISM) was produced as a copy of the core part of the SEON View with no modifications (such as in the Quality Assurance initiative).

Table 6.3. Software Testing Initiative: Structural Mappings.

Core Ontology Concept	ISO 29110 Element	ISO 12207 Element	ISO 29119 Element
Performed Process (event)	Process (event) Activity (event)	Process (event) Activity (event)	Process (event)
Performed Activity (event)	Activity (event) Task (event)	Process (event) Activity (event) Task (event)	Process (event) Activity (event) Task (event)
Artifact (object)	Product (object)	Artifact (object)	Information Item (object) Document (object)
Stakeholder (agent)	Role (agent)	Stakeholder (agent)	Stakeholder (agent)

The structural harmonization of three standards from the same standardization organization (ISO) could be seen as a simple task; however, it was not the case. Supporting the Henderson-Sellers et al. (2014) concerns, standards like these present several inconsistences and divergences, even in their structures. Some of them are perceived in Table 6.3 (for details, see (RUY et al., 2017b)). The vocabulary applied for naming the elements is diverse. For example, four distinct terms are used for Artifacts in the three standards. ISO/IEC 29119, although defining the terms *Information Item* and *Document*, its application causes confusion in some sections. Stakeholders also receive a different designation, *Role*, in ISO/IEC 29110. Another divergence refers to the work units' granularity. Although the standards use the same structure of work units (being Processes composed of Activities that are composed of Tasks), the granularity varies significantly. In an extreme case, work units represented as Processes in ISO/IEC 29119 are represented as Tasks in ISO/IEC 29110. Other problems regarding Artifacts, Stakeholders and Outcomes descriptions were also detected during this structural harmonization effort.

Proceeding to the Content Harmonization phase, a content model (SCM) was built for each standard, supported by the SSMs. These models were mapped to the SEON View, achieving a coverage of 63% for ISO/IEC 12207, 63% for ISO/IEC 29110, and 45% for ISO/IEC 29119, the most detailed standard. Finally, an integrated content model (ICM) was established considering the addition of new activities and artifacts.

During the modeling and mapping activities other issues regarding the standards' contents were identified: (i) the testing contents are organized in different ways (in specific processes or dispersed in activities and tasks); (ii) the standards' presentations varies in terms, abstraction level, writing style, and sometime omit relevant information; (iii) differences in granularity; and (iv) use of terms. Although most of the listed issues do not constitute an error in the standards, they certainly affect understanding, and any initiative aiming at using them together. Additionally, this initiative highlighted the different ways the ISO standards are organized and represented. For instance, ISO/IEC 29119 is much more detailed than the other two standards, many times going beyond what is expected for such standards, describing "what to do", and also providing "how to do" the processes.

The Software Testing ICM included new elements representing notions not covered in the vertical mappings. The low coverage (63%, 63% and 45%) was caused by two main reasons:

(i) the SCMs contain several standards' elements collapsing testing and other notions out of the initiative scope (such as *Software Integration and Tests*, from ISO/IEC 29110; and *Integrate the software units and software components and test it,* from ISO/IEC 12207). These elements were considered only partially covered;

(ii) SEON does not represent aspects related to testing initial parameters (e.g., feature sets, test conditions and test coverage items), test procedures, and testing specifications. In fact, this initiative showed us that SEON does not cover some important testing aspects, such as those related to test procedures, which need to be represented in the next releases of SEON (a feedback for SEON evolution).

This study was important for evaluating Harmony with a different, more experienced profile. The participant's experience is very high in the testing domain, high in conceptual modeling and medium regarding the considered standards. This is the user profile we expect to use Harmony from the standards' users' side (such as in SPI projects in software organizations). Thus, the initiative was conducted more independently and reached more consolidated results. At the end, we have applied a questionnaire and an interview with similar questions to those used in the second experiment (see Appendix B). The obtained feedback confirmed some of the findings we had in the previous experiment, and provided us with some new insights on Harmony.

Regarding the difficulties faced, the participant reported that was ***easy*** to use the Approach, the Tool and the Ontologies (she knew very well the ontologies used in this initiative due to her previous participation in their construction). The Conceptual Models were reported with ***medium*** degree of difficulty; and the Standards remained as the ***hard*** aspect. The main problems reported were related to: (i) dealing with a large amount of information; (ii) the need to be precise for matching the right notions (often subjective in the standards); (iii) the presence of elements collapsing two or more notions (mainly in activities from ISO/IEC 29110 and ISO/IEC 12207); and (iv) the standards' descriptions, when presenting ambiguity, implicit information, lack of uniformity, and distinct description perspectives.

Similarly to Study 2, the participant considered that the provided Ontologies, Approach and Tool had a ***high*** or ***very high*** level of contribution for the harmonization initiative. The reported advantages (only those not already presented in the previous experiment) are described in the following. The participant said the **Approach** contributed by: (i) providing proper activities for dealing with the standards' structure and contents, and for comparing and integrating them; (ii) supporting the definition of the content models by specializing the structural models (easier to understand, faster to do and with better results); (iii) offering a holistic view of the standards (structural and content models), easing their analysis, mappings and integration; (iv) requiring/motivating a semantic analysis of the standards elements; (v) allowing a better understanding of the standards' specificities (even for experts); (vi) improving the precision, consistency and general quality of the results; (vii) providing results (mappings and integrated models) valuable for conducting other works (e.g. harmonization of other standards, standard-to-standard mappings, harmonized processes). Finally, the main consideration about **Mapper** was the provision of a visual and productive mapping interface, where the notions could be selected directly from the diagrams, allowing more contextualized and consistent matches.

6.4 Final Considerations

We have conducted three empirical studies for evaluating Harmony throughout its incremental development. Although we had already some ideas inspired in the literature and our previous experience, these evaluations were essential for improving the approach and making explicit some aspects germane for this work.

The initial experiments (Studies 1 and 2) were used as basis for defining some directions for the approach, such as modeling the standards' structure and contents, developing a supporting tool and defining how to better explore semantic aspects. The Study 3 provided us a deeper analysis of the approach guidelines and results. It has confirmed many expectations of Harmony development and previous experiments, such as those related to Harmony's ability to guide the users, efforts employed, use of ontologies and conceptual models, reuse potential, precision of the models and mappings, semantic orientation, and quality of results. Besides the discussed results and feedback reports, the set of evaluations (all studies) has also shown that the approach can support different user profiles. The ideal profile for applying Harmony is high experienced professionals, mainly in the SE domain and in the involved standards (such as standards' developers, for creating or maintain standards). However, we also expect the approach could be used by professionals with an intermediary experience with standards; for example, aiming at applying an integrated view of some standards for conducting SPI initiatives in software organizations (standards' users).

Harmony applications were crucial for evolving it to the version presented in Chapter 5. Several findings and insights were applied for improving and refining it, and for planning future work (as discussed in the next chapter). Moreover, the results produced in many of these initiatives (standards' models, mappings, integrated models, new ICM elements and standards' problems) are a rich material for future projects involving new harmonization initiatives, SEON improvements, definition of harmonized processes, ontological analysis of standards, and so on.

Finally, along all the work, we have identified diverse standards' inconsistences and incompatibilities (such as those in (RUY et al., 2017)). It strengthens our initial motivations that the available SE standards are not properly aligned, and that there is a semantic interoperability problem to be addressed. It also reinforces our claims that the development and use of a harmonization approach like Harmony is a need for dealing with the current standards and for improving the new versions, alleviating semantic problems in the future.

7 Final Considerations

This final chapter presents the main conclusions regarding the research presented in this thesis. In Section 7.1, a brief summary points out the main aspects presented along the text. Section 7.2 describes the contributions achieved, relating them to the proposed objectives and to the published papers. Section 7.3 discusses the limitations of the work. Finally, Section 7.4 presents the future perspectives concerning improvements of ideas discussed in this thesis, and development of new research initiatives from the basis established by this work.

7.1 Summary of the Research

This research addresses the harmonization of Software Engineering (SE) process-related standards, a semantic interoperability problem. Along the last decade, several works have addressed this problem, proposing a number of approaches. In a **Systematic Mapping of the Literature**, we have identified and analyzed studies addressing the topic, producing a panorama of the current research status. These works applied a set of techniques, relying on different knowledge models and focusing on varied SE standards. They provided a number of contributions for the topic, with valuable research advances. However, semantic treatment is still in an initial stage. We identified research gaps on how these approaches deal with the semantic underlying the standards. Two main aspects were selected to be further explored in this thesis. Firstly, standards harmonization heavily depends on a consistent domain knowledge representation, supporting the identification of similarities between standards, and offering a semantic endowed solution for dealing with the divergences. Secondly, it requires an approach to systematically conduct the harmonization efforts, encouraging an effective semantic judgment supported by the domain knowledge.

For fulfilling the domain knowledge representation needs, we have created **SEON**, an Ontology Network for the SE domain. It is composed of well-founded and modular ontologies representing diverse SE subdomains. SEON is not a final artifact or a complete SE ontology. Contrariwise, it is an effort for establishing the structure, mechanisms and initial content for a long-term evolution work, aiming at being a useful and consistent source of SE domain knowledge. Currently, the network comprises a total of 13 ontologies and more than 300 concepts regarding nine SE subdomains. SEON has been successfully applied in our empirical evaluations conducting harmonization initiatives, where it is pointed as an essential artifact for supporting semantic aware decisions.

Concerning the methodological aspects, we have developed **Harmony**, an ontology-based approach for harmonizing SE standards. Harmony explores commonly applied harmonization techniques, combining them in a systematic process highly aligned to SEON. Semantic aspects are expected to be explored along the process. One of its central ideas is representing the standards' information as conceptual models, favoring comparisons and integration. The final result is an integrated model, representing a unified view of the involved standards. This integrated model is based on SEON's ontologies and adapted with and linked to the standards' information. Finally, the Mapper tool supports Harmony use, providing mapping features and assuring information integrity.

Some **Empirical Evaluations** were conducted for assessing Harmony (and indirectly SEON). They complete the DSR *Design Cycle*, being crucial for the research development, as the feedback provided was used for improving and refining Harmony and Mapper, as well as SEON's contents, mechanisms and presentation. Moreover, these studies provided us with a practical view of the approach and network behavior, and helped to identify and understand some research limitations and perspectives for future work.

7.2 Research Contributions

This work is located between two areas: Software Engineering (SE) and Ontology Engineering (OE). The general objective – *to develop an ontology-based approach for SE standards harmonization* – is clearly addressing a SE problem: the lack of alignment of SE standards. This issue mainly affects software organizations conducting SPI initiatives wherein distinct standards are to be adopted in combination. A considerable amount of knowledge on the SE domain (including diverse subdomains), on software processes and on SE standards have been applied in several research steps for achieving the expected contributions.

However, although the research ends are for SE, the means have an expressive portion of OE. In order to reach the discussed results, knowledge on ontologies, ontology networks, and ontology patterns was systematically applied (and sometimes even produced). Hence, we see this work as an application of OE for a SE problem, with contributions in both areas.

The central contributions of this research are directly related to the development of Harmony and SEON, regarding the research steps for reaching them, the process and artifact themselves, and the results of the evaluations conducted. More specifically, our main contributions are:

- **A Systematic Mapping on SE Standards Harmonization**. The mapping has focused on studies considering a common model for supporting harmonization efforts, and pointed out an increased interest in the topic. We have also identified the main areas, standards, techniques and types of common models used and the focuses given by the researches. The mapping revealed a panorama of the topic, which can be a valuable material for future harmonization research efforts. In the context of this thesis, we highlight the findings about the ways the techniques are evolving, the use of ontologies and the research gaps. These points were essential for determining our main directions regarding our knowledge framework (SEON) and harmonization approach (Harmony). A manuscript describing the mapping, "Harmonizing Software Engineering Standards Using Common Conceptual Models: a Mapping Study", was submitted to the *Software Quality Journal (SQJ)* in last June and is under review.

- **The Software Engineering Ontology Network**. SEON was conceived as our SE knowledge framework to be used as a semantic reference in harmonization efforts. We have applied the ideas of Ontology Networks (SUÁREZ-FIGUEROA et al., 2012) and established SEON from previously developed ontologies we have evolved and integrated (such as SPO (BRINGUENTE et al., 2011) (FALBO et al., 2013), CMPO (CALHAU; FALBO, 2012), RSMO (BARCELLOS et al., 2010) and ROoST (SOUZA et al., 2013)), and new ontologies we have produced for enlarging the network coverage (such as RDPO, DPO, CPO and QAPO). SE ontologies have been produced by our group for several years. SEON is a way to integrate them in an evolving framework, assuring the network premises, and maximizing their application in diverse research initiatives. The first version of SEON was published in the *proceedings of 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW'16)* (RUY et al., 2016). Since then, SEON doubled the size. The specification of its current version is available at dev.nemo.inf.ufes.br/seon/. Some aspects related to SEON are contributions by themselves, namely:
    - **SEON's Architecture**: We defined a layered architecture, based on ontology generality levels, which we believe applies to ontology networks in general. This architecture organizes the ontologies in the network and empowers the ontology derivation chain (from foundational to core to domain ontologies), providing grounding, encouraging reuse, and easing the addition and integration of new ontologies.
    - **Building and Integration Mechanisms**: Pattern-oriented development is a promising solution in OE (PRESUTTI et al., 2009) (FALBO et al., 2013); however, when applied in an ON, especially using OPLs, it has shown more interesting results regarding quality and productivity. In an ON of a single domain (e.g., SE), the same patterns are reused more frequently, due to the common basis and similarities between the subdomains. Our efforts to build SEON helped us to improve the OPLs and their patterns, and to integrate the ontologies (as they share a similar structure given by analogous modeling decisions). Guidelines for ontology integration were also defined, aiding to identify the integration context and to define our strategy.
    - **Body of Ontologies**: SEON has currently SPO as core ontology (plus two external core ontologies) and 12 domain ontologies for nine subdomains, totalizing around 320 concepts (not counting UFO). All these concepts are grounded, defined, and contextualized in their respective networked ontologies. It is a significant body of knowledge, which can be applied

in diverse contexts (such as tool development and integration, knowledge management, semantic documentation, and standards harmonization).

- **SEON Specification**: The specification built from the SEON models is a facility for accessing, understanding and analyzing the network. It solves a recurrent problem of publishing our models in a fast way, while provides complete information for the interested people. It has been used as reference in several works performed at NEMO, as well as for disclosing our work on SE ontologies to other research groups. We have also identified several improvement opportunities from the specification, with respect to definitions, relations and network organization. According to Google Analytics, in the last year, it has been frequently accessed, for more than 400 users from diverse countries, such as Brazil (43% of the accesses), Russia (22%), USA (11%), Germany (6%), Spain (4%) and others (14%).

- **Research Path towards SEON**. For building the network, we have experienced varied approaches to reach some essential characteristics (such as a robust SE ground and effective building mechanisms). These efforts produced some artifacts and leaded us to an understanding of a better way to build the network. The most important are:
  - **SEMDM Ontological Analysis**: By analyzing SEMDM, the ISO/IEC 24744 metamodel, we have found some problems and proposed solutions for them. This task helped us to improve our models and to provide a better alignment to the ISO conceptualization. Moreover, the rationale for identifying problems is being reused for detecting similar problems / inconsistences in more specific standards (such as concepts overlap in ISO/IEC 12207 and product classification in CMMI-DEV). This ontological analysis was published in the *proceedings of the 8th International Conference on Formal Ontology in Information Systems (FOIS'14)* (RUY et al., 2014).
  - **ISP-OPL**: Building this OPL has advanced our research regarding three main points: establishing an OPL on software process, later being used as basis for improving SP-OPL (and SEON); collecting valuable data on the experiment applying an OPL for building domain ontologies; understanding some essential practical requirements for a harmonization approach (such as using grounded ontologies and improving the mapping techniques). This OPL was published in the *proceedings of the 30th Annual ACM Symposium on Applied Computing (SAC'2015)* (RUY et al., 2015), where we received the *Best Paper Award* in the category "AI and Agents", and an invitation for submitting an extended paper. This extended paper included an experiment, and was published in the *ACM SIGAPP Applied Computing Review Journal* (RUY et al., 2015b).
  - **Ontology Patterns**: For building our ontologies, among the OE methods, we have focused on the reuse-based ones. The experiences dealing with ontology patterns and OPLs since the

beginning of this research have led us to a more mature understanding of this topic, resulting in more effective OPLs and improvements in the way to build and apply them as presented in the SEON guidelines. We believe we have some interesting advances also in this topic, since it was a fertile subject for investigation. Initially, we have published an Enterprise OPL in the *proceedings of the 29th Annual ACM Symposium on Applied Computing (SAC'2014)* (FALBO et al., 2014). Then, the two aforementioned papers about ISP-OPL. These results and other from our research group resulted in the chapter "Ontology Pattern Languages" in the *Ontology Engineering with Ontology Design Patterns: Foundations and Applications* book (FALBO et al., 2016). Finally, from some ideas we have elaborated from the joint application of Domain-Related and Foundational Ontology Patterns, we published the paper "Ontology Engineering by Combining Ontology Patterns" in the *proceedings of the 34th International Conference on Conceptual Modeling (ER'2015)* (RUY et al., 2015c). This paper led to a journal invitation and we produced an extended version, including approaches for deriving and applying ontology patterns, published in the *Data & Knowledge Engineering (DKE)* journal (RUY et al., 2017).

  - **SPO / SP-OPL Improvements**: Along this path, diverse improvements were introduced in SPO, making it the main ontology in SEON, and the main source of SE knowledge for grounding the domain networked ontologies. Now we are working on a new version of SP-OPL, considering the discussed advances.

- **The Ontology-based Standards Harmonization Approach**. Harmony is a harmonization approach focusing on dealing with standards' semantics. It advances the research on standards harmonization by reducing the gaps identified in the systematic literature mapping. Compared to relevant researches in this topic (such as (FERCHICHI et al., 2008) (JENERS et al., 2013) (PARDO et al., 2013) (HENDERSON-SELLERS et al., 2014)), some Harmony's contributions can be listed. It defines a process offering detailed instructions and practical considerations to be performed by users with different profiles. Three important harmonization techniques were combined and adjusted for dealing with the semantics of the standards' structure and contents. The knowledge referential comes from SEON, a consistent source of SE domain knowledge. The structural harmonization performed as part of the process (and not a fixed embedded solution) gives some flexibility, allowing the user to focus on the more relevant aspects of the standards according to the initiative purpose and scope. At the same time, it supports the onwards content harmonization. The mappings and integration from conceptual models, grounded in UFO, also have contributed to the accomplishment of the initiatives and the consistency of the obtained results. The match types allow for more precise and informative links during the mappings, enriching the established connections. The resulting integrated model carries an ontological source and the matches established with the involved standards, a valuable information that can be used in other initiatives. Finally, Mapper supports the approach, allowing the user to focus on the semantic decisions. We believe it is a significant material for publication. At the moment, the Chapter 5 of this thesis is being adapted to a manuscript to be submitted for the *Computer Standards & Interfaces* Journal.

- **Works Applying Harmony:** Along this research, we have applied Harmony in a number of initiatives, producing reports containing the mappings and integrated models for several SE subdomains. Although some of them may require reviews and improvements, the results on Quality Assurance, Software Measurement, Dynamic Testing and Configuration Management are quite interesting and can be further explored in new researches. For example, the integrated models of the three first domains were applied in a recent experiment (in 2017/1) for supporting the definition of harmonized software processes. Moreover, the initiative for harmonizing testing processes of three ISO standards (ISO/IEC 12207, ISO/IEC 29110 and ISO/IEC 29119) was used as basis for publishing the paper "Software Testing Processes in ISO Standards: How to Harmonize Them?" in the *proceedings of the XVI Brazilian Symposium on Software Quality (SBQS'2017)* (RUY et al., 2017b). We expect to discuss also the other results in other publications.

These results contribute to fulfill the established objectives of this work. Table 7.1 relates the presented contributions to the specific objectives of this thesis.

Table 7.1. Contributions versus Specific Objectives.

General Objective	Specific Objective	Contribution
Develop an ontology-based approach for SE standards harmonization	**S01.** Perform a systematic literature mapping for analyzing related studies	Systematic Mapping of the Literature on SE Standards Harmonization
	SO2. Establish a SE reference knowledge framework	The Software Engineering Ontology Network . *SEON Architecture* . *Building and Integration Mechanisms* . *Body of Ontologies* . *SEON Specification* Research path towards SEON . *SEMDM Ontological Analysis* . *ISP-OPL* . *Guidelines to derive and use Ontology Patterns* . *SPO / SP-OPL Improvements*
	SO3. Develop a SE standards harmonization approach grounded in SEON	Harmony, an Ontology-based Standards Harmonization Approach Works Applying Harmony

7.3 Research Limitations

The development of this research has involved a significant effort regarding diverse aspects in two main knowledge areas (OE and SE). Diverse previous works were used for supporting, guiding, inspiring and composing our solutions. Besides the literature mapping, we have explored two main lines in this thesis resulting in the development of an ontology network and a harmonization approach. Both are properly documented and preliminarily evaluated. However, these proposals are new and we are aware of the limitations they present. Although some specific limitations are already discussed along the previous chapters, this section points out the main ones concerning the complete work.

Regarding the **Systematic Mapping of the Literature**, besides the typical limitations of secondary studies, involving papers selection, classification and analysis, we detach a scope limitation. The mapping was conducted considering publications addressing harmonization of SE standards by means of common models. This "<u>common models</u>" restriction is important in the context of this thesis because it is a key point in our solution as we decided to work with a reference knowledge framework (SEON). However, when seen as a panorama of the standards harmonization topic, the mapping is limited. We have disregarded some papers presenting techniques and solutions for mapping and integrating standards without using a common model. We believe it is not a serious flaw since the most mature works (PARDO et al., 2013) (JENERS et al., 2013) (HENDERSON-SELLERS et al., 2014) (PARDO et al., 2015) recognize the need for a knowledge support to address this semantic interoperability problem.

SEON is an ontology network designed to grow and evolve in a long-term work. This research only established the initial features and included in it some domain ontologies. Although the current version is already able to be used in some contexts, the following aspects can be pointed out as limitations:

- <u>Patterns Support</u>: it is an important mechanism for creating domain ontologies. However, although the main patterns have been well established (e.g. those related to performed processes in SP-OPL), some of them need to be updated and properly documented.
- <u>Integration Mechanisms</u>: we have provided a set of instructions on how to integrate ontologies in SEON. However, ontology integration is a complex activity requiring a further research and a more detailed guidance.
- <u>SEON Coverage</u>: currently, SEON covers some aspects of nine SE subdomains. It is a good starting point; though, along the time, ontologies on other SE disciplines should be developed and integrated to the network.
- <u>Contributions to SEON</u>: nowadays, SEON is being developed exclusively by our group. Although some external ontologies can be reused and incorporated to the network, it is not enough. New mechanisms allowing other researchers to contribute to SEON (suggesting, proposing or adding new ontologies as well as evaluating the existing ones) are needed.

Harmony is result of studies on standards harmonization approaches and how conceptual models (mainly ontologies) can support it. It was proposed and applied in a proof-of-concept and three empirical evaluations. These applications were fundamental for improving the approach and also to identify some limitations, as presented below:

- Focus on SE Process-related Standards: Harmony was developed considering only SE standards, more specifically, *software process-related standards*. Although we believe it can be applied, with some adjustments, in a wider context, we have not considered other situations in our empirical evaluations.
- Complexity of the Harmonization Activity: harmonizing multiple standards developed with different backgrounds is a complex activity. It encompasses efforts on understanding the subject domain and the involved standards, and on identifying each piece of related information, linking the similarities and solving the divergences. It is not a simple task and, besides methods and tools, demands experience and abilities. Thus, we must be aware of the complex nature of the standards harmonization activity. It should be clear that our proposal does not completely solve the harmonization problem, but intends to advance the topic by offering a sound support for the involved efforts.
- Dependence on the Users' Profile: during a harmonization initiative, a number of decisions shall be taken considering diverse factors. The Harmony process provides guidelines for the most common situations, and Mapper supports some decisions with information and consistency checking. However, a successful initiative highly depends on the users' judgment and decisions, mainly for dealing with the standards' semantics. Although the approach provides some guidance, just following it does not guarantee satisfactory results. The user should have a good experience in the target domain and also in the involved standards. Besides that, due to Harmony's model-oriented approach, the user should be familiar with building and analyzing conceptual models. Although ontologies could be seen as a complex aspect, the approach only requires their understanding, and the empirical studies have not pointed out any relevant difficulty in this respect. Hence, for Harmony application, it is expected some experience in the involved domain and standards, and also at least a basic experience in analyzing and building conceptual models.
- Tool Support: Mapper offers a partial support for performing Harmony. It focuses on the content mapping and integration activities. Mapper could provide a better support if it covers all the activities and produces the complete report, allowing the user to perform a larger portion of the process supported by the tool.
- Harmonization of Relations: Harmony's core unit is the element. Information from the standards is modeled in a concise element, and then it is analyzed, mapped and integrated. Relations between these elements (when provided) are also modeled and used for supporting the

elements analysis. However, they are important information that is not formally mapped or integrated. In the current approach, for example, if the activity *Planning* _produces_, _changes_, _evaluates_ or _uses_ a certain *Document*, only *Planning* and *Document* will be effectively mapped and integrated. The relations are not systematically analyzed or compared. Relation harmonization is not included in Harmony (and neither in any other analyzed approach) to not increase the complexity of the solution. However, it is an open issue to be considered in the future.

- Real World Evaluation: Harmony was applied in a proof-of-concept, and three empirical studies. A total of nine initiatives have been conducted considering seven SE subdomains and seven distinct software process standards. A good range of the SE domain was covered and some of the main software process standards used (e.g. CMMI-DEV, ISO/IEC 12207, ISO/IEC 29110 and MR-MPS-SW). Also users with different profiles applied the approach, from lower to higher experienced ones. However, all the initiatives occurred in academic contexts. A real word project, for example, for harmonizing standards aiming at deploying them in a software organization, would be a worthy context for effectively evaluating Harmony.

7.4 Perspectives of Future Works

The results presented in this thesis establish the basis for a number of future works. We believe these works can solve some of the aforementioned limitations and others are improvements for the work done and contributions in the context of the research lines explored in this thesis. Following, we present a set of possible future works, encompassing both OE and SE areas, and, in particular, the Standards Harmonization topic.

While designing and developing SEON, we have worked diverse aspects on **Ontology Engineering** mainly regarding ontology development, ontology patterns and ontology networks. These aspects can be further explored, both in a general way and for SEON improvements, as discussed below:

- Ontology Networks: we believe the architecture proposed for SEON, taking the ontology generality levels and building / integration mechanisms into account, can be applied for creating other ontology networks. Besides the discussed features, with the network growth, aspects such as maintainability and consistency assurance become more important. Thus, new mechanisms can be included, such as better use of metadata, versioning, support for axioms definition and reuse, other types of consistency check, and new forms of visualization.
- Ontology Pattern Languages: we have experienced the potential of ontology patterns, organized as OPLs, for building domain ontologies. Important advances can be made in this topic. Some perspectives for OPLs are: (i) providing a sound specification, considering the process models and DROPs details, such as exemplified in (FALBO et al., 2016) and in the NEMO's OPL

project[12]; (ii) making this specification available in a navigable format, similar to the SEON Specification (actually, the SEON specification tool can be adapted for that); (iii) developing an effective tool for specifying and applying DROPs, improving the version presented in (RUY et al., 2017); and (iv) including FOPs as support for DROPs application as proposed in (RUY et al., 2017).

- Ontology Integration: the integration mechanisms can be further explored. We have provided some guidelines for SEON, but it should offer more detailed guidance for ontology engineers, considering ontology grounding, similarity between concepts, and integration scope. Works on deep semantics and ontology alignment and matching (such as (EUZENAT; LE DUC, 2012)) can help. Moreover, we believe diverse of the ideas we are using for harmonizing standards, by mapping and integrating models using varied match types, could be applied to improve ontology integration.
- SEON Specification: it can evolve for admitting comments and suggestions, enhanced search, links to external references (such as papers and other ontologies), and visualization improvements.
- SEON Patterns: SP-OPL should be extended to completely cover the current version of SPO. Moreover, the general improvements suggested for OPLs also apply for SP-OPL.
- SEON Coverage: the body of ontologies composing SEON shall be increased for allowing a better coverage of the SE domain. Ontologies covering other subdomains should be developed and added to SEON (e.g. *Documentation*, *Maintenance*, *Portfolio Management*, *Risks Management*). Some already modeled subdomains (e.g. *Construction*, *Project Management*, *Software Measurement*) should be increased, and some aspects should be further explored (such as those related to *human resources*, *artifacts*, *procedures*, and *software tools*).
- SEON Application: SEON has a specific purpose within this thesis; however, it has a sound potential working as a SE foundation for punctual solutions (development of specific tools, basis for semantic annotations, definition of specific processes), as well as a SE integrated knowledge reference to be applied in semantic interoperability problems (such as those related to knowledge management, semantic documentation, and tools / services integration). Improvements in the operational version (currently in OWL) can also open a new line of applications for SEON in a Semantic Web based approach for SE (ZHAO et al., 2009), such as publishing reusable software engineering knowledge resources and providing services for searching, reasoning and querying. The more SEON is applied, more feedback can be collected for improving it.

[12] https://nemo.inf.ufes.br/projects/opl/

Along Harmony development, diverse ideas related to **Standards Harmonization** have been applied and some remained as future perspectives.

- Harmony Process Improvements: from the empirical studies, some improvements were identified for Harmony process. The approach could: (i) consider instructions for dealing with more situations, such as those for overcoming problems in standards' descriptions, such as differences in abstraction levels, implicit information and inconsistencies; (ii) better address elements partially out of scope; and (iii) provide guidelines for reusing previous models and mappings. Moreover, additional activities for defining harmonized processes or describing an integrated standard could be further provided.
- Harmony Process Adaptations: the process could be adapted for other contexts. Harmony is applied for harmonizing SE process standards. However, we believe it can be adapted (or give rise to new versions) for considering other types of standards, such as Software Product, IT Governance and System Engineering, or even for dealing with processes in other areas, such as Engineering, Telecommunication, Manufacturing etc. Working with models, which can represent a variety of information sources, and a set of match types gives some flexibility to the solution for dealing with other contexts. Going beyond, promising subjects for harmonization / integration using a Harmony-like approach are system integration, regulations and laws, ontologies and so on.
- Mapper: the Harmony's supporting tool can be improved in many directions: (i) offering support for the entire process, better covering the structural mapping and integration activities; (ii) providing more detailed information for supporting matching decisions; (iii) creating a library of models to be reused, in particular standards' structural and content models built in previous initiatives; (iv) reusing also matches from previous initiatives; (v) improving the coverage calculation (e.g., allowing elements weight and composite matches in horizontal mappings); and (vi) usability improvements related to model reloading, elements splitting, model navigation and reports produced.
- Real World Applications: Harmony should be applied in real projects. It can be employed in SPI initiatives, for providing to the organization a unified view of standards to be deployed (supporting planning and conducting an improvement project), or for mapping the organization's standard process to an intended standard (supporting the gap analysis). In these contexts, aspects such as operationality, adaptability to new situations, needed efforts and effectiveness of the results could be better assessed. It could also be used by standards' developers, for producing standards' mappings (as MR-MPS-SW and ISO/IEC 29110 do), for providing a consensual model of related standards (as the work described in (HENDERSON-SELLERS et al., 2014) intends to do), or for identifying inconsistencies to be solved in

standards' reviews. All these applications can result in valuable feedback for improving Harmony.

Other future perspectives can also be discussed considering researches in **Software Engineering**.

- Process Definition: the integrated model resultant from a harmonization initiative (the ICM) represents a unified view of the involved standards. It can be used for diverse ends. For example, for defining a harmonized process. It can be done taking the ICM as basis for manually describing the process, or transforming the ICM in a process model enriched with the process-specific information, able to be automatically transformed to a process description.
- Tools Development: SEON provides models describing SE subdomains. Harmony's integrated models also describe portions of the SE domain, aligned to standards. These models are valuable sources for the development of SE tools. Besides that, SEON as a network, also provides these models in an integrated mode, being useful for the development of integrated environments and for tool / service integration initiatives.
- Feedback for Standards: besides the mappings and integrated models produced, harmonization initiatives usually identify inconsistencies and divergences in the involved standards. This material can be useful for improving the available SE standards.
- Ontology-based SE Learning: we believe the body of ontologies composing SEON can be applied for SE learning. Many complex disciplines (such as Project Management, Software Measurement and Configuration Management) can be better understood with the support of consolidated conceptual models as ontologies are.

References

AAEN, I. Software process improvement: Blueprints versus recipes. *IEEE Software*, 20:86–93, 2003.

ALMEIDA, J.P.A., CARDOSO, E.C.S., GUIZZARDI, G. On the Goal Domain in the RM-ODP Enterprise Language: An Initial Appraisal based on a Foundational Ontology. In 6th Workshop on ODP for Enterprise Computing (WODPEC'2010), *Proc. of the 14th IEEE Int. Enterprise Distributed Object Computing Conference Workshops (EDOCW'2010)*, IEEE Computer Society Press, 382-390, 2010.

BALDASSARRE, M.T., PIATTINI, M., PINO, F.J., VISAGGIO, G. Comparing ISO/IEC 12207 and CMMI-DEV: towards a mapping of ISO/IEC 15504-7, *ICSE'2009, WOSQ Workshop*. Vancouver, 2009.

BALDASSARRE, M.T., CAIVANO, D., PINO, F.J., PIATTINI, M. and VISAGGIO, G. Harmonization of ISO/IEC 9001:2000 and CMMI-DEV: from a theoretical comparison to a real case application. *Software Quality Journal*, 20(2), pp.309-335, 2012. [13]

BANHESSE, E.L., SALVIANO, C.F. and JINO, M. Towards a metamodel for integrating multiple models for process improvement. In *38th Euromicro Conference on Software Engineering and Advanced Applications*. pp. 315-318. IEEE, 2012.

BARCELLOS, M.P., FALBO, R.A., MORO, R.D. A Well-Founded Software Measurement Ontology. In: *6th International Conference on Formal Ontology in Information Systems*, v.209. p.213–226, Toronto, 2010.

BARCELLOS, M.P. and FALBO, R.A. A software measurement task ontology. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. pp. 311-318. ACM, 2013.

BARCELLOS, M.P., FALBO, R.A. and FRAUCHES, V.G.V. Towards a Measurement Ontology Pattern Language. In *1st Joint Workshop Onto.Com/ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering*, Rio de Janeiro, 2014.

BIFFL, S., WINKLER, D., HÖHN, R. and WETZEL, H. Software process improvement in Europe: potential of the new V-modell XT and research issues. *Software Process: Improvement and Practice*, 11(3), pp.229-238, 2006.

BORST, W.N. *Construction of engineering ontologies for knowledge sharing and reuse*. University of Twente, 1997.

BOURQUE, P. and FAIRLEY, R.E. *Guide to the Software Engineering Body of Knowledge (SWEBOK (R))*: Version 3.0. IEEE Computer Society Press, 2014.

BRINGUENTE, A.C., FALBO, R.A. and GUIZZARDI, G. Using a Foundational Ontology for Reengineering a Software Process Ontology. *Journal of Information and Data Management*, v. 2, n. 3, pp.511, 2011.

BUSCHMANN, F., HENNEY, K. and SCHMIDT, D.C. *Pattern-Oriented Software Architecture: On Patterns and Pattern Languages*, John Wiley & Sons Ltd, 2007.

CALERO, C., RUIZ, F. and PIATTINI, M. (editors). *Ontologies for Software Engineering and Software Technology*. Springer Science & Business Media, 2006.

CALHAU, R.F. and FALBO, R.A. An Ontology-based Approach for Semantic Integration. In: *14th IEEE International Enterprise Distributed Object Computing Conference*, Vitória, Brazil. Los Alamitos: IEEE Computer Society, pp.111-120, 2010.

CALHAU, R.F. and FALBO, R.A. A Configuration Management task ontology for semantic integration. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pp.348-353, ACM, 2012.

[13] References in blue are those considered in the Systematic Mapping of the Literature.

CARVALHO, V.A. and ALMEIDA, J.P.A. Toward a well-founded theory for multi-level conceptual modeling. *Software & Systems Modeling*, pp.1-27, 2016.

CMMI INSTITUTE. *CMMI Maturity Profile Report*, 2016-2. Available online on http://partners.cmmiinstitute.com/cmmi-appraisals/process-maturity-profiles. 2016.

D'AQUIN, M., SCHLICHT, A., STUCKENSCHMIDT, H. and SABOU, M. Criteria and Evaluation for Ontology Modularization Techniques, In: *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, Springer-Verlag, pp. 67–89, 2009.

D'AQUIN, M. and GANGEMI, A. Is there beauty in ontologies? *Applied Ontology*, vol. 6, n.3, p.165–175, 2011.

D'AQUIN, M. Modularizing Ontologies, In: Suárez-Figueroa, M. C., et al. (editors), *Ontology Engineering in a Networked World*. Springer, Berlin, 2012.

DAVENPORT, T.H. The coming commoditization of processes. *Harvard Business Review*, 83(6), pp.100-108, 2005.

DUARTE, B.B., SOUZA, V.E.S., LEAL, A.L.C., FALBO, R.A., GUIZZARDI, G., GUIZZARDI, R.S.S. Towards an Ontology of Requirements at Runtime. *Proc. of the 9th International Conference on Formal Ontology in Information Systems*, Annecy, France, 2016.

DUARTE, B.B., LEAL, A.L.C., FALBO, R.A., GUIZZARDI, G., GUIZZARDI, R.S.S., SOUZA, V.E.S. Ontological foundations for software requirements with a focus on requirements at runtime. *Applied Ontology Journal*, v.13, pp.73-105, 2018.

DUARTE, K.C. and FALBO, R.A. Uma ontologia de qualidade de software. (in Portuguese). In *Workshop de Qualidade de Software,* João Pessoa, pp.275-285, 2000.

EESSAAR, E. and SGIRKA, R. An Ontological Analysis of Metamodeling Languages. In *Information Systems Development*. pp. 381-392, Springer New York, 2013.

ERIKSSON, H. The Semantic-Document Approach to Combining Documents and Ontologies. *International Journal of Human-Computer Studies*, v.65, n.7, pp.624-639, 2007.

ESPINDOLA, R.S. Integração de Padrões de Qualidade na Área de Melhoria de Processo de Software (in Portuguese). Doctorate Thesis, PUC-RS, Porto Alegre, 2011.

EUZENAT, J. and LE DUC, C. Methodological guidelines for matching ontologies. In *Ontology engineering in a networked world*. pp.257-278, Springer Berlin Heidelberg, 2012.

FALBO, R.A. Integração de Conhecimento em um Ambiente de Desenvolvimento de Software (in Portuguese). Doctorate Thesis. Federal University of Rio de Janeiro, Rio de Janeiro, 1998.

FALBO, R.A., RUY, F.B. and DAL MORO, R. Using Ontologies to Add Semantics to a Software Engineering Environment. In *International Conference on Software Engineering and Knowledge Engineering (SEKE'2005)*, pp. 151-156, 2005.

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* 4(4), pp.239-250, 2009.

FALBO, R.A., BARCELLOS, M.P., NARDI, J.C., GUIZZARDI, G. Organizing Ontology Design Patterns as Ontology Pattern Languages. In: *Proceedings of the 10th Extended Semantic Web Conference (ESWC'2013)*. Montpellier, France, 2013.

FALBO, R.A., GUIZZARDI, G., GANGEMI, A. and PRESUTTI, V. Ontology patterns: clarifying concepts and terminology". In *Proceedings of the 4th Workshop on Ontology and Semantic Web Patterns*, Sydney, 2013b.

FALBO, R.A., RUY, F.B., GUIZZARDI, G., BARCELLOS, M.P., ALMEIDA, J.P.A. Towards an Enterprise Ontology Pattern Language. In: *Proc. of the 29th Annual ACM Symposium on Applied Computing (SAC '2014)*, pp. 323–330, 2014.

FALBO, R.A., BARCELLOS, M.P., RUY, F.B., GUIZZARDI, G., GUIZZARDI, R.S.S. Ontology Pattern Languages. In Gangemi, A., Hitzler, P., Janowicz, K., Krisnadhi, A., and Presutti, V., editors, *Ontology Engineering with Ontology Design Patterns: Foundations and Applications*. IOS Press, 2016.

FERCHICHI, A., BOUREY, J.P., BIGAND, M. and LEFEBVRE, H. Implementing integration of quality standards CMMI and ISO 9001:2000 for software engineering. In *Complex Systems Concurrent Engineering*, Springer London, pp.133-142, 2007.

FERCHICHI, A., BIGAND, M., and LEFEBVRE, H. An ontology for quality standards integration in software collaborative projects. In *Proceedings of the First International Workshop on Model Driven Interoperability for Sustainable Information Systems (MDISIS'08)*, pages 17–30, Montpellier, France, 2008.

FERREIRA, A.L., MACHADO, R.J., PAULK, M.C., Size and Complexity Attributes for Multimodel Improvement Framework Taxonomy. *36th EUROMICRO Conference on Software Engineering and Advanced Applications*, 2010.

FETTKE, P. AND LOOS, P. Ontological Evaluation of Reference Models Using the Bunge-Wand-Weber Model. *Proceedings of the 2003 Americas Conference on Information Systems*, Florida, USA, 2003.

FUGGETTA, A. Software process: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering*, pp. 25-34. ACM, 2000.

GANGEMI, A., PISANELLI, D.M. and STEVE, G. An Overview of the ONIONS Project: Applying Ontologies to the Integration of Medical Terminologies. *Data Knowledge Engineering*, 31, 2, pp. 183–220, 1999.

GANGEMI, A., GUARINO, N., MASOLO, C., OLTRAMARI, A. and SCHNEIDER, L. Sweetening ontologies with DOLCE. In *International Conference on Knowledge Engineering and Knowledge Management*. pp.166-181, Springer Berlin Heidelberg, 2002.

GARCÍA, F., BERTOA, M., CALERO C., VALLECILLO, A., RUÍZ, F., PIATTINI, M., and GENERO, M. Towards a consistent terminology for software measurement. *Information and Software Technology* 48 (8), pp.631-644, 2006.

GONZALEZ-PEREZ, C. and HENDERSON-SELLERS, B. An Ontology for Software Development Methodologies and Endeavours. In: *Ontologies for Software Engineering and Software Technology*. Springer Science & Business Media, 2006.

GONZALEZ-PEREZ, C., HENDERSON-SELLERS, B., MCBRIDE, T., LOW, G.C. and LARRUCEA, X. An Ontology for ISO software engineering standards: 2) Proof of concept and application. *Computer Standards & Interfaces*, 48, pp.112-123, 2016.

GRUBER, T.R. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), pp.199-220, 1993.

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

GUARINO, N. Formal Ontology and Information Systems. In *Proceeding of the International Conference on Formal Ontology in Information Systems (FOIS'98)*, vol. 46, pp. 3–15, 1998.

GUARINO, N., OBERLE, D. and STAAB, S. What is an Ontology?. In *Handbook on ontologies*. pp.1-17, Springer Berlin Heidelberg, 2009.

GUIZZARDI, G. *Ontological Foundations for Structural Conceptual Models*. Fundamental Research Series. Centre for Telematics and Information Technology. Netherlands, 2005.

GUIZZARDI, G. On Ontology, ontologies, Conceptualizations, Modeling Languages and (Meta)Models, In: Vasilecas, O., Edler, J., Caplinskas, A. (edtors) *Databases and Information Systems IV*, pp.18-39, IOS Press, Amsterdam, 2007.

GUIZZARDI, G., FALBO, R.A., 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 XI Iberoamerican Workshop on Requirements Engineering and Software Environments*, pp. 244-251. Recife, Brazil, 2008.

GUIZZARDI, G., WAGNER, G., FALBO, R.A., GUIZZARDI, R.S.S, ALMEIDA, J.P.A. Towards Ontological Foundations for the Conceptual Modeling of Events. In: *32th International Conference on Conceptual Modeling (ER'2013)*, pp.327-341. Hong-Kong, China, 2013.

GUIZZARDI, G., WAGNER, G., ALMEIDA, J.P.A., GUIZZARDI, R.S.S. Towards Ontological Foundation for Conceptual Modeling: The Unified Foundational Ontology (UFO) Story. *Applied Ontology*, Vol. 10, issues 3-4, IOS Press, 2015.

HELLER, B., HERRE, H. Ontological Categories in GOL. *Axiomathes* 14: 71-90, Kluwer Academic Publishers, 2004.

HENDERSON-SELLERS, B. Standards harmonization: theory and practice. *Software & Systems Modeling*, 11(2), pp.153-161, 2012.

HENDERSON-SELLERS, B., GONZALEZ-PEREZ, C., MCBRIDE, T. and LOW, G. An ontology for ISO software engineering standards: 1) Creating the infrastructure. *Computer Standards & Interfaces*, 36(3), pp.563-576, 2014.

HEVNER, A.R., MARCH, S.T., PARK, J. and Ram, S. Design science in information systems research. *MIS quarterly*, 28(1), pp.75-105, 2004.

HEVNER, A.R. A three cycle view of design science research. *Scandinavian journal of information systems*, 19(2), p.4, 2007.

ISO, ISO 9001. Quality Management Systems – Requirements, 2015.

ISO/IEC, ISO/IEC 12207. Systems and Software Engineering – Software Life Cycle Processes, 2008.

ISO/IEC, ISO/IEC 15288. Systems and Software Engineering – System Life Cycle Processes, 2008b.

ISO/IEC, ISO/IEC 15504. Information Technology – Process Assessment, 2012.

ISO/IEC, ISO/IEC 15939. Systems and Software Engineering – Measurement Process, 2007b.

ISO/IEC, ISO/IEC 20000. Information Technology – Service Management, 2011b.

ISO/IEC, ISO/IEC 24744. Software Engineering – Metamodel for Development Methodologies, 2007.

ISO/IEC, ISO/IEC 25010. Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – System and Software Quality Models, 2011b.

ISO/IEC, ISO/IEC 27001. Information Technology – Security Techniques – Information Security Management Systems – Requirements, 2013.

ISO/IEC, ISO/IEC 27002. Information Technology – Security Techniques – Code of Practice for Information Security Controls, 2013b.

ISO/IEC, ISO/IEC TR 29110. Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs), 2011.

ISO/IEC, ISO/IEC 29119. Software and Systems Engineering - Software Testing (Parts 1, 2, 3 and 4), 2013c.

ISO. Standards published by ISO/IEC JTC1 / SC7. Available on http://www.jtc1-sc7.org/. Accessed in April, 2013.

ISO/IEC, ISO/IEC JTC 1 / SC 7 Software and Systems Engineering. Available on https://www.iso.org/committee/45086.html. Accessed in September, 2017.

ITGI. Aligning COBIT 4.1, ITIL V3 and ISO/IEC 27002 for Business Benefit. IT Governance Institute (ITGI) and Office of Government Commerce (OGC), 131, 2008.

JALALI, S. and WOHLIN, C. Systematic literature studies: database searches vs. backward snowballing. In *International Symposium on Empirical Software Engineering and Measurement*, 29–38, 2012.

JENERS, S., LICHTER, H. and DRAGOMIR, A. Towards an integration of multiple process improvement reference models based on automated concept extraction. In *European Conference on Software Process Improvement*. Springer Berlin Heidelberg, pp. 205-216, 2012.

JENERS, S., LICHTER, H. and PYATKOVA, E. Metric based Comparison of Reference Models based on Similarity. *International Journal of Digital Content Technology and its Applications*, 6(21), p.50, 2012b.

JENERS, S., LICHTER, H. and ROSENKRANZ, C.G. Efficient Adoption and Assessment of Multiple Process Improvement Reference Models. *e-Informatica Software Engineering Journal*, 7(1), 2013.

JENERS, S. and LICHTER, H. Smart integration of process improvement reference models based on an automated comparison approach. In *European Conference on Software Process Improvement*. pp. 143-154. Springer Berlin Heidelberg, 2013.

KELEMEN, Z.D., KUSTERS, R., TRIENEKENS, J. Identifying criteria for multimodel software process improvement solutions - based on a review of current problems and initiatives. *Journal of Software: Evolution and Process*, 24(8), pp.895-909, 2012.

KITCHENHAM, B.A. and CHARTERS, S. Guidelines for performing Systematic Literature Reviews in Software Engineering, School of Computer Science and Mathematics, Keele University and Department of Computer Science University of Durham, UK, v. 2.3, 2007.

KITCHENHAM, B.A., BUDGEN, D. and BRERETON, O.P. Using mapping studies as the basis for further research – a participant-observer case study. *Information and Software Technology*, 53(6), pp.638-651, 2011.

LARRUCEA, X. and SANTAMARIA, I. An industrial assessment for a multimodel framework. *Journal of Software: Evolution and Process*, 26(9), pp.837-845, 2014.

LEPASAAR, M., and MÄKINEN, T. Integrating software process assessment models using a process meta model. In *Engineering Management Conference*. IEMC'02. IEEE International, vol. 1, pp. 224-229, 2002.

LIAO, L., QU, Y. and LEUNG, H. A software process ontology and its application. In *Proceedings of the First International Workshop on Semantic Web Enabled Software Engineering*, 2005.

LOCHMANN, K. and GOEB, A. A unifying model for software quality. In *Proceedings of the 8th International Workshop on Software Quality*, pp. 3-10. ACM, 2011.

MALONE, J., BROWN, A., LISTER, A.L., ISON, J., HULL, D., PARKINSON, H., and STEVENS, R. The Software Ontology (SWO): a resource for reproducibility in biomedical data analysis, curation and digital preservation. *Journal of Biomedical Semantics*, 5, 25, 2014.

MÄKINEN, T. and VARKOI, T. A harmonized design for process assessment indicators. *Software Process: Improvement and Practice* 12 (4), pp.331-338, 2007.

MCBRIDE, T., HENDERSON-SELLERS B., and reviewers. The growing need for alignment, ISO/IEC JTC1/SC7 N5507, May 2012.

MEJIA, J., MUÑOZ, E. and MUÑOZ, M. Reinforcing the applicability of multi-model environments for software process improvement using knowledge management. *Science of Computer Programming*, 121, pp.3-15, 2016.

MENDES O. and ABRAN, A. Software engineering ontology: A development methodology, *Metrics News*, vol. 9, pp. 68-76, 2004.

MENDES, O. and ABRAN, A. Issues in the Development of an Ontology for an Emerging Engineering Discipline. *First Workshop on Ontology, Conceptualizations and Epistemology for Software and Systems Eng. (ONTOSE)*. Alcalá Henares, Spain, 2005.

MUTAFELIJA, B. and STROMBER, H. ISO 9001:2000 - CMMI V1.1 Mappings. *Software Engineering Institute*, Boston, USA, Tech. Rep. V1.1., Dec. 2003.

NARDI, J.C. and FALBO, R.A. Uma Ontologia de Requisitos de Software. (in Portuguese). In *Ibero-American Conference on Software Engineering (CIbSE'2006)*, pp.111-124, 2006.

NEGRI, P.P., SOUZA, V.E.S., LEAL, A.L.C., FALBO, R.A. and GUIZZARDI, G. Towards an Ontology of Goal-Oriented Requirements. *Proc. of the 20th Iberoamerican Conference on Software Engineering (CIbSE'2017)*, Buenos Aires, 2017.

OATES, B. J. *Researching Information Systems and Computing*. Sage, 2011.

OBERLE, D., GRIMM, S. and STAAB, S. An ontology for software. In *Handbook on Ontologies*, 2nd ed. Berlin: Springer-Verlag, 2009.

OBRST, L. Ontological Architectures. In: Poli, R., Healy, M., Kameas, A. (editors), *Theory and Applications of Ontology: Computer Applications*, Springer, 2010.

O'LEARY, D. Using AI in Knowledge Management: Knowledge Bases and Ontologies, *IEEE Intelligent Systems*, vol. 13, pp.34-39, 1998.

PARDO, C., PINO, F.J., GARCÍA, F., VELTHIUS, M.P. and BALDASSARRE, M.T. Trends in harmonization of multiple reference models. In *International Conference on Evaluation of Novel Approaches to Software Engineering*. Springer Berlin Heidelberg. pp. 61-73, 2010.

PARDO, C., PINO, F., GARCÍA, F., ROMERO, F.R., PIATTINI, M. and BALDASSARRE, M.T. HProcessTOOL: a support tool in the harmonization of multiple reference models. In *International Conference on Computational Science and Its Applications*. Springer Berlin Heidelberg. pp.370-382, 2011.

PARDO, C., PINO, F.J., GARCÍA, F., PIATTINI, M. and BALDASSARRE, M.T. An ontology for the harmonization of multiple standards and models. *Computer Standards & Interfaces*, 34(1), pp.48-59, 2012.

PARDO, C. *A framework to support the harmonization between multiple models and standards*. PhD Tesis, Universidad de Castilla-La Mancha, Ciudad Real, Spain, 2012.

PARDO, C., PINO, F.J., GARCÍA, F., BALDASSARRE, M.T. and 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), pp.125-143, 2013.

PARDO, C., GARCÍA, F., PIATTINI, M., PINO, F.J. and BALDASSARRE, M.T. A reference ontology for harmonizing process-reference models. *Revista Facultad de Ingeniería Universidad de Antioquia*, (73), pp.29-42, 2014.

PARDO. C., GARCÍA, F., PIATTINI, M., PINO, F.J. and BALDASSARRE, M.T. A 360-degree process improvement approach based on multiple models. *Revista Facultad de Ingeniería Universidad de Antioquia*, (77), pp.95-104, 2015.

PARDO, C., GARCÍA, F., PIATTINI, M., PINO, F.J., LEMUS S. and BALDASSARRE, M.T. Integrating Multiple Models for Definition of IT Governance Model for Banking ITGSM. *International Business Management*, 10, pp.4644-4652, 2016.

PARDO, C., PINO, F.J. and GARCÍA, F. Towards an Integrated Management System (IMS), harmonizing the ISO/IEC 27001 and ISO/IEC 20000-2 Standards. *International Journal of Software Engineering and Its Applications*, 10(9), pp.217-230, 2016b.

PARK, J. and RAM, S. Information Systems Interoperability: What lies Beneath?, *ACM Transactions on Information Systems*, 22, 595-632, 2004.

PAULK, M.C. Comparing ISO 9001 and the capability maturity model for software. *Software Quality Journal*, 2(4), pp.245-256, 1993.

PETERSEN, K., FELDT, R., MUJTABA, S. and MATTSSON, M. Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, 17(1), 2008.

PETERSEN, K., VAKKALANKA, S. and KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64, pp.1-18, 2015.

PINO, F.J., GARCÍA, F. and PIATTINI, M. Software process improvement in small and medium software enterprises: a systematic review. *Software Quality Journal*, 16(2), pp.237-261, 2008.

PRESUTTI, V., DAGA, E., GANGEMI, A. and BLOMQVIST, E. eXtreme Design with Content Ontology Design Patterns. In *Proceedings of the Workshop of Ontology Patterns (WOP'09)*, 2009.

QUIRINO, G.K.S., BARCELLOS, M.P. and FALBO, R.A. OPL-ML: A Modeling Language for Representing Ontology Pattern Languages. *5th International Workshop on Ontologies and Conceptual Modeling (OntoCom'17)*, 2017.

RAHMAN, A.A., SAHIBUDDIN, S. and IBRAHIM, S. Using taxonomy comparative analysis for the unification of process improvement frameworks. *International Journal of Digital Content Technology and its Applications*, 6(21), p.34, 2012.

ROUT, T. SPICE and the CMM: is the CMM compatible with ISO/IEC 15504?. *AquIS'98*, p.12, 1998.

ROUT, T., TUFFLEY, A. and CAHILL, B. *Capability Maturity Model Integration Mapping to ISO/IEC TR 15504-2:1998*. Software Quality Institute - Griffith University, Australia, 2001.

ROUT, T.P. and TUFFLEY, A. Harmonizing ISO/IEC 15504 and CMMI. *Software Process: Improvement and Practice*, 12(4), pp.361-371, 2007.

RUY, F.B., FALBO, R.A., BARCELLOS, M.P., GUIZZARDI, G. An Ontological Analysis of the ISO/IEC 24744 Metamodel. In *Proceedings of the 8th International Conference on Formal Ontology in Information Systems (FOIS'14)*, Rio de Janeiro, Brazil, 2014.

RUY, F.B., FALBO, R.A., BARCELLOS, M.P. and GUIZZARDI, G. Towards an ontology pattern language for harmonizing software process related ISO standards. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing (SAC'2015)*. pp.388-395. ACM, 2015.

RUY, F.B., FALBO, R.A., BARCELLOS, M.P., GUIZZARDI, G. and QUIRINO, G.K. An ISO-based software process ontology pattern language and its application for harmonizing standards. *ACM SIGAPP Applied Computing Review*, 15(2), pp.27-40, 2015b.

RUY, F.B., REGINATO, C.C., SANTOS, V.A., FALBO, R.A. and GUIZZARDI, G. Ontology engineering by combining ontology patterns. In *Proceedings of the 34th International Conference on Conceptual Modeling (ER'2015)*. pp.173-186. Springer International Publishing, 2015c.

RUY, F.B., FALBO, R.A., BARCELLOS, M.P., COSTA, S.D., GUIZZARDI, G. SEON: A Software Engineering Ontology Network. In *Proceedings of 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW'16)*, Bologna, Italy, pp.527-542, 2016.

RUY, F.B., REGINATO, C.C., SANTOS, V.A., FALBO, R.A. and GUIZZARDI, G., From Reference Ontologies to Ontology Patterns and Back. Accepted for publication in *Data & Knowledge Engineering* (DKE), 2017.

RUY, F.B., SOUZA, E.F., FALBO, R.A., BARCELLOS, M.P. Software Testing Processes in ISO Standards: How to Harmonize Them? In Proceedings of the XVI Brazilian Symposium on Software Quality (SBQS'2017), Rio de Janeiro, Brazil, 2017b.

SALVIANO, C.F. and FIGUEIREDO, A.M.C. Unified Basic Concepts for Process Capability Models. In *International Conference on Software Engineering and Knowledge Engineering (SEKE'2008)*, pp. 173-178, 2008.

SCHERP, A., SAATHOFF, C., FRANZ, T. and STAAB, S. Designing Core Ontologies. *Applied Ontology*, v. 6, n.3, pp.177–221, 2011.

SEI, CMMI for Development, Version 1.3. CMU/SEI-2010-TR-033. Software Engineering Institute, Carnegie Mellon University, 2010.

SEI, *The Process Improvement in Multimodel Environments (PrIME) project*, 2010b. Available at http://www.sei.cmu.edu/process/research/prime-details.cfm.

SHANKS, G., TANSLEY, E. and WEBER, R. Using Ontology to Validate Conceptual Models. *Communications of the ACM* 46, pp.85-89, 2003.

SICILIA, M.A., CUADRADO, J.J., GARCÍA, E., RODRÍGUEZ, D., HILERA, J.R. The Evaluation of Ontological Representation of the SWEBOK as a Revision Tool. In: *29th International Computer Software and Application Conference (COMPSAC)*, pp. 26-28, Edinburgh, UK, 2005.

SIVIY, J., KIRWAN, P., MARINO, L., and MORLEY, J. The value of harmonizing multiple improvement technologies: A process improvement professional's view. Technical report, Software Engineering Institute, Carnegie Mellon, 2008.

SMITH, B. Against Idiosyncrasy in Ontology Development. *Frontiers in Artificial Intelligence and Applications*, 150, p.15, 2006.

SOFTEX, *Brazilian Software Process Improvement (MPS.BR) - General Guide: 2016*, Brazil, 2016.

SOFTEX, Association for Promotion of Brazilian Software Excellence. Brazilian Software Process Improvement (MPS.BR). Available on http://www.softex.br/mpsbr/. Accessed in September, 2017.

SOMMERVILLE, I. *Software Engineering*. International Computer Science Series. Ed. Addison Wesley, 2004.

SOUZA, E.F., FALBO, R.A. and VIJAYKUMAR, N.L. Using Ontology Patterns for Building a Reference Software Testing Ontology. In: *17th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW)*, pp.21-30, Vancouver, 2013.

SOUZA, E.F., FALBO, R.A. and VIJAYKUMAR, N.L. ROoST: Reference Ontology on Software Testing. *Applied Ontology*, v.12, pp.59-90, 2017.

SOYDAN, G. and KOKAR, M. An OWL Ontology for Representing the CMMI-SW Model. In *2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE)*, Athens, Greece, pp. 1-14, 2006.

STUDER, R., BENJAMINS, V.R. and FENSEL, D. Knowledge engineering: principles and methods. *Data & Knowledge Engineering*, 25(1-2), pp.161-197, 1998.

SUÁREZ-FIGUEROA, M.C., GÓMEZ-PÉREZ, A., MOTTA, E. and GANGEMI, A. *Ontology Engineering in a Networked World*. Springer Science & Business Media, 2012.

VAN LAMSWEERDE, A. Goal-oriented requirements engineering: A guided tour. *Proc. of the 5th IEEE International Symposium on Requirements Engineering*. pp. 249. IEEE Computer Society, 2001.

VELTMAN, K.H. Syntactic and semantic interoperability: new approaches to knowledge and the semantic web. *New Review of Information Networking*, 7(1), pp.159-183, 2001.

WANG, X., GUARINO, N., GUIZZARDI, G. and MYLOPOULOS, J. Towards an Ontology of Software: a Requirements Engineering Perspective. *Proc. of the 8th International Conference on Formal Ontology in Information Systems*, Rio de Janeiro, Brazil, v.267, pp.317–329, 2014.

WIERINGA, R., MAIDEN, N., MEAD, N. and ROLLAND, C. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, 11(1), pp.102-107, 2006.

WOHLIN, C., RUNESON, P., NETO, P.A.M.S., ENGSTRÖM, E., MACHADO, I.C. and ALMEIDA, E.S. On the reliability of mapping studies in software engineering. *Journal of Systems and Software*, 86(10), pp.2594-2610, 2013.

WONGTHONGTHAM, P., CHANG, E., DILLON, T., SOMMERVILLE, I. Development of a Software Engineering Ontology for Multisite Software Development. *IEEE Transactions on Knowledge and Data Engineering*, v. 21, n. 8, pp.1205-1217, 2009.

WÜRSCH, M., GHEZZI, G., HERT, M., REIF, G. and GALL, H.C. SEON: A Pyramid of Ontologies for Software Evolution and its Applications. *Computing*, v. 94, n. 11, pp.857-885, 2012.

YOO, C., YOON, J., LEE, B., LEE, C., LEE, J., HYUN, S. and WU, C. A unified model for the implementation of both ISO 9001:2000 and CMMI by ISO-certified organizations. *Journal of Systems and Software* 79 (7), pp. 954–961, 2006.

ZHAO, Y., DONG, J. and PENG, T. Ontology Classification for Semantic-Web-Based Software Engineering. *IEEE Transactions on Services Computing*, 2 (4), October-December, pp.303-317, 2009.

Appendix A

Harmonizing Quality Assurance Processes

This appendix presents the report of an initiative harmonizing the Quality Assurance processes of CMMI-DEV and ISO/IEC 12207. It is a proof-of-concept of Harmony produced by the doctorate candidate and review by the advisors. The results are organized according to the Harmony activities.

A.1. Purpose and Scope Definition

A.1.1. Identify the Harmonization Purpose

Provide a Harmonized Model for the Quality Assurance Processes of CMMI-DEV and ISO/IEC 12207.

A1.2. Define the Harmonization Scope

Software Process and Product Quality Assurance, from Planning to Evaluation and Noncompliance Control, including processes, their activities, related artifacts and involved stakeholders.

A1.3. Select the Standards' Portions

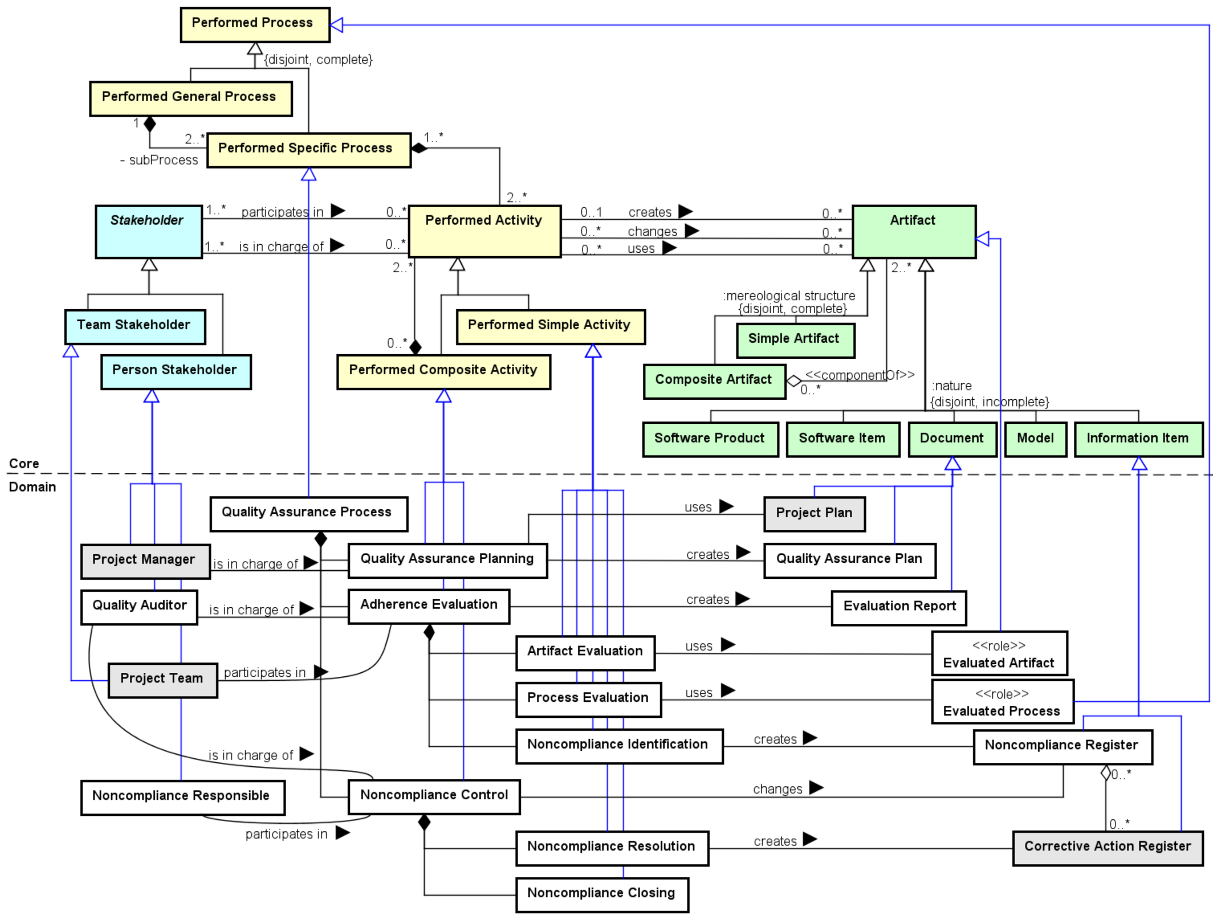
CMMI-DEV (v.1.3): PPQA Process Area, and related generic practices applied to PPQA, namely: GP 2.2 - Plan the (Quality Assurance) Process, GP 2.4. - Assign Responsibility and GP 2.7 - Identify and Involve the Relevant Stakeholders.

ISO/IEC 12207 (2008): the whole Software Quality Assurance Process (7.2.3), and specific activities from other two processes for dealing with nonconformities control, namely: Quality management corrective action (6.2.5.3.2) from Quality Management Process (6.2.5), and Problem resolution (7.2.8.3.2) from Software Process Resolution Process (7.2.8).

A1.4. Select the SEON View

SPO (core ontology): portion dealing with processes, activities, stakeholders and artifacts.

QAPO (domain ontology), the whole ontology, as shown in the diagram below.

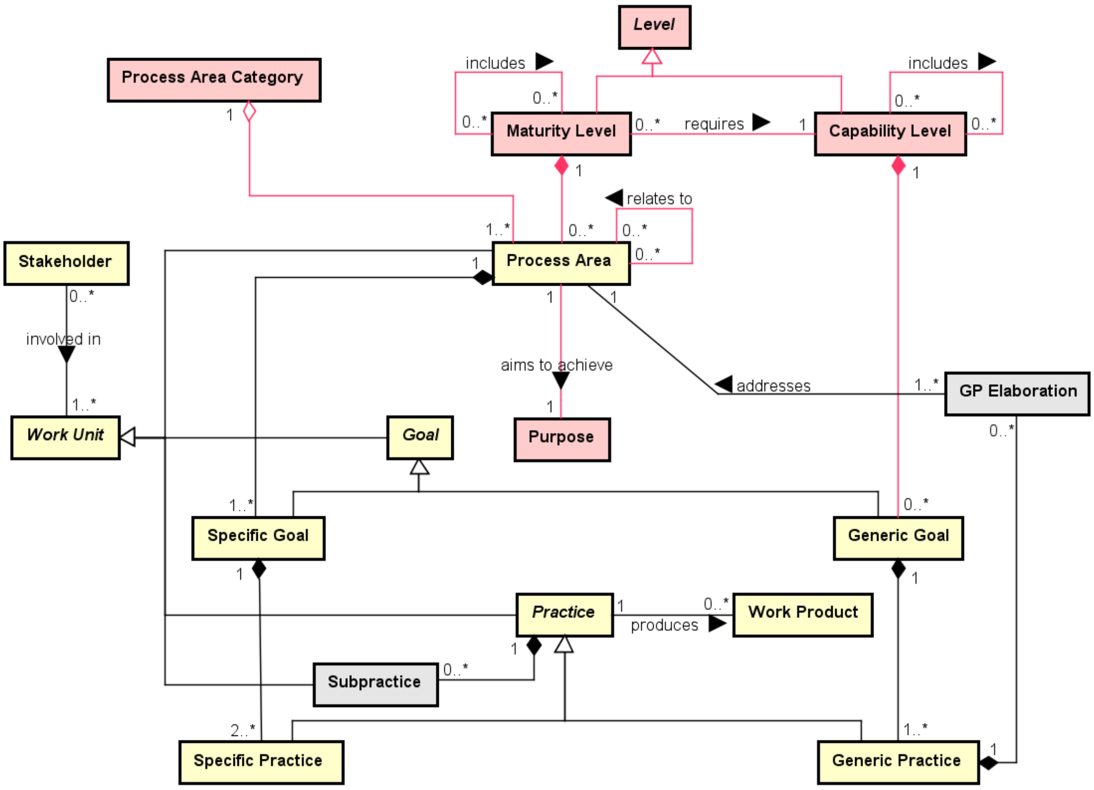


A.2. Structural Harmonization

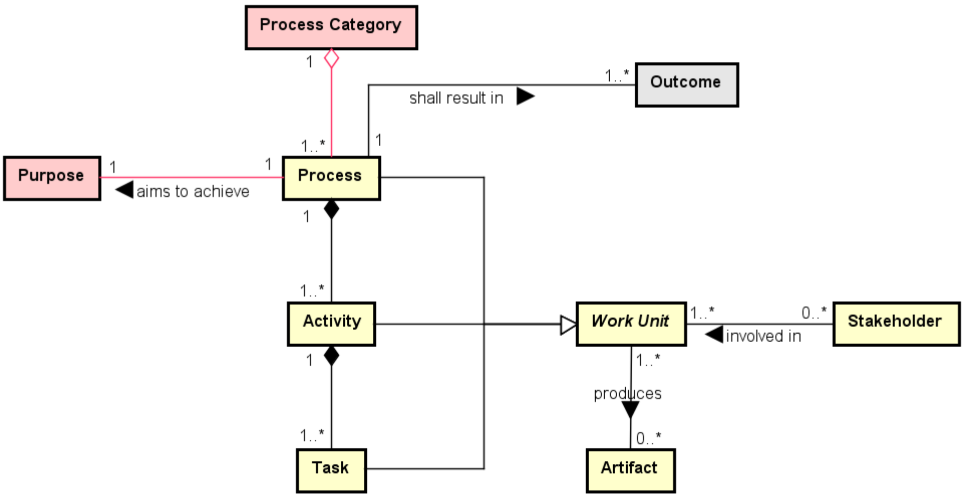
A2.1. Develop the Standards' Structural Models

Two Standards' Structural Models (SSMs) were developed. Elements in yellow and relations in black are those considered for being extended in the initiative (according to the defined scope); elements and relations in pink were discarded; elements in gray are used only as information source.

CMMI SSM



ISO/IEC 12207 SSM



A2.2. Map Standards' Structural Models to Core Ontologies

The Core Ontologies' concepts and standards elements are classified according to the ontological types (*event*, *object*, *agent* and *moment*). The standards' elements considered are compared to the core concepts with the same ontological type and matches are established between them (vertical mappings), as shown in the table.

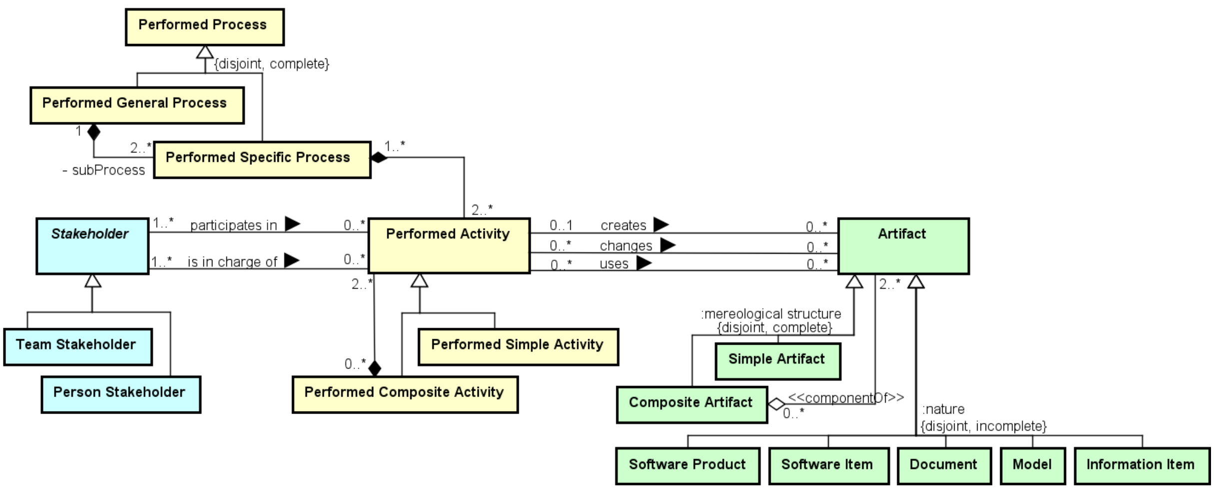
Core Ontology Concept	CMMI-DEV Element	ISO/IEC 12207 Element
Performed Process (event)	Process Area (event) Specific Goal* (event) Generic Goal (event)	Process (event) Activity (event)
Performed Activity (event)	Specific Goal (event) Generic Goal (event) Specific Practice (event) Generic Practice (event)	Process (event) Activity (event) Task (event)**
Artifact (object)	Work Product (object)	Artifact (object)
Stakeholder (agent)	Stakeholder (agent)	Stakeholder (agent)

* Although the Goal term could be referred as a moment (the goal of a process), in the CMMI-DEV structure, it is used as a part of a process area and has practices as its parts. Thus, in this harmonization initiative, the CMMI-DEV's Generic / Specific Goals are considered events.

** Task is an element considered only when it denotes an event. Otherwise it will be used as an element for information.

A2.3. Develop the Integrated Structural Model

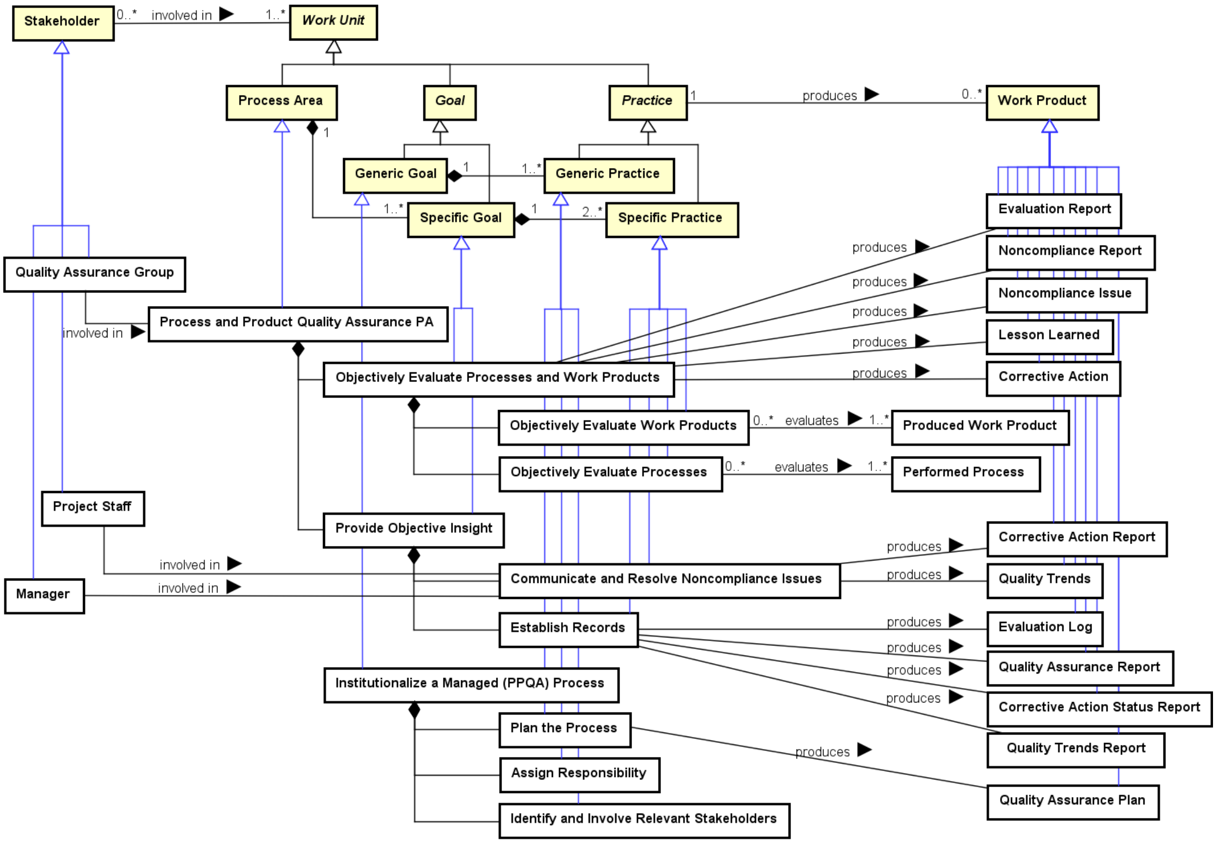
All the SSM elements were covered by the core concepts of the SEON View. Thus, there are no modifications to be made in the ISM and it corresponds to the core portion of the SEON View, as shown below.



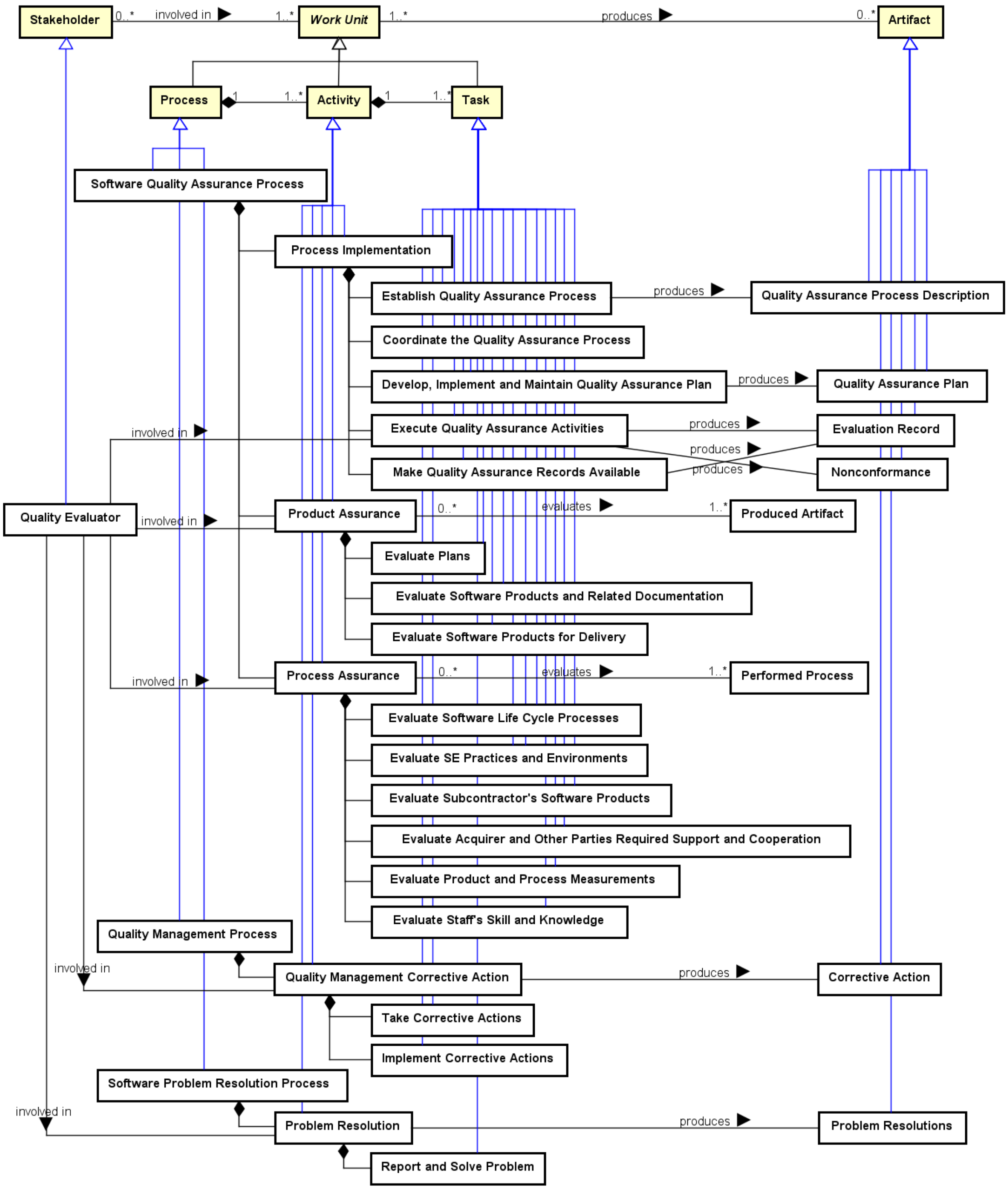
A.3. Content Harmonization

A3.1. Develop the Standards' Content Models

CMMI-DEV SCM: created by specializing the CMMI-DEV SSM considering the defined standard portion.



ISO/IEC 12207 SCM: created by specializing the ISO/IEC 12207 SSM considering the defined standard portion.



A3.2. Map Standards' Content Models to Domain Ontologies

Background colors in the first column points out the elements coverage: green for fully covered; yellow for partially covered; blank for not covered. Composite matches are shown with the target (third column) including two or more concepts.

Content Vertical Mapping: CMMI-DEV ⇨ SEON View (Coverage: 61%[14])

CMMI-DEV Element	Match	SEON View Concept
EVENTS		
Process and Product Quality Assurance PA	EQUIVALENT	Quality Assurance Process
Objectively Evaluate Processes and Work Products	EQUIVALENT	Adherence Evaluation
Objectively Evaluate Work Products	WIDER {*Noncompliance identification not covered*}	Artifact Evaluation
	WIDER {*Artifact evaluation not covered*}	Noncompliance Identification
	EQUIVALENT	(Artifact Evaluation + Noncompliance Identification)
Performed Process	ACT AS	Evaluated Process
Objectively Evaluate Processes	WIDER {*Noncompliance identification not covered*}	Process Evaluation
	WIDER {*Process evaluation not covered*}	Noncompliance Identification
	EQUIVALENT	(Process Evaluation + Noncompliance Identification)
Provide Objective Insight	WIDER {*Communication not covered*}	Noncompliance Control
Communicate and Resolve Noncompliance Issues	WIDER {*Communication not covered*}	Noncompliance Resolution
	WIDER {*Communication and resolution actions not covered*}	Noncompliance Closing
Establish Records	NORELATION	
Institutionalize a Managed (PPQA) Process	OVERLAP {*Considers institutionalization actions not included in the process, but it does not have the main activities (evaluations and control)*}	Quality Assurance Process
Plan the Process	PARTIAL	Quality Assurance Planning
Assign Responsibility	PARTIAL	Quality Assurance Planning
Identify and Involve Relevant Stakeholders	OVERLAP {*Involvement only planned not performed*}	Quality Assurance Planning
	OVERLAP {*Involvement only performed, not planned*}	Adherence Evaluation
	OVERLAP {*Stakeholders communication not covered*}	Noncompliance Control
OBJECTS		
Evaluation Report	EQUIVALENT	Evaluation Report
Noncompliance Report	NORELATION	
Noncompliance Issue	EQUIVALENT	Noncompliance Register

[14] **Coverage** is an approximated number, calculated with a basic algorithm, considering the coverage done by the matches established for each standard element. An element coverage can be 0% (no matches), 33% (partially covered), 67% (largely covered), 0.67% to 90% (multiple partial matches) and 100% (fully covered). The mapping coverage ranges from 0% to 100% and is given by the sum of each individual element coverage divided by the number of elements (Σec/Σe). It does not take into account complexity or types of the elements, just the set coverage and quantity.

CMMI-DEV Element	Match	SEON View Concept
Lesson Learned	NORELATION	
Corrective Action	EQUIVALENT	Corrective Action Register
Produced Work Product	ACT AS	Evaluated Artifact
Corrective Action Report	NORELATION	
Quality Trends	NORELATION	
Evaluation Log	NORELATION	
Quality Assurance Report	NORELATION	
Corrective Action Status Report	NORELATION	
Quality Trends Report	NORELATION	
Quality Assurance Plan	EQUIVALENT	Quality Assurance Plan
AGENTS		
Quality Assurance Group	WIDER {*Documentation and reporting responsibilities not covered*}	Quality Auditor
Project Staff	EQUIVALENT	Project Team
Manager	EQUIVALENT	Project Manager

Mapping Analysis: the CMMI-DEV PPQA process area is covered in **61%** by the SEON View. Most of the uncovered portion regards documentation and communication activities and the related artifacts and stakeholders.

Content Vertical Mapping: ISO/IEC 12207 ⇨ SEON View (Coverage: 90%)

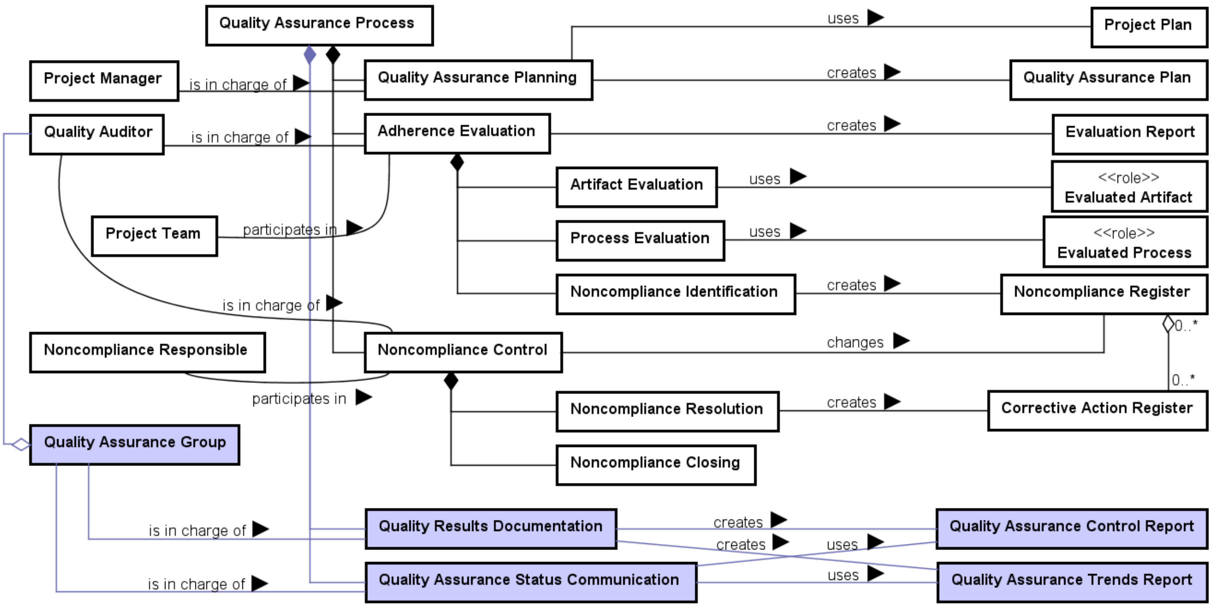
ISO/IEC 12207 Element	Match	SEON View Concept
EVENTS		
Software Quality Assurance Process	PARTIAL {*Needs the other processes (for managing nonconformities and corrective actions) to be complete*}	Quality Assurance Process
Process Implementation	PARTIAL	Quality Assurance Process
Establish Quality Assurance Process	PARTIAL	Quality Assurance Planning
Develop, Implement and Maintain Quality Assurance Plan	EQUIVALENT	Quality Assurance Planning
Execute Quality Assurance Activities	PARTIAL {*Execute Quality Assurance Activities does not objectively evaluate*}	Adherence Evaluation
Make Quality Assurance Records Available	NORELATION	
Product Assurance	PARTIAL {*Product Assurance does not objectively evaluate*}	Artifact Evaluation
Evaluate Plans	PARTIAL	Artifact Evaluation
Evaluate Software Products and Related Documentation	PARTIAL	Artifact Evaluation
Evaluate Software Products for Delivery	PARTIAL	Artifact Evaluation
Performed Process	ACTS AS	Evaluated Process
Process Assurance	PARTIAL {*Process Assurance does not objectively evaluate*}	Process Evaluation
Evaluate Software Life Cycle Processes	PARTIAL	Process Evaluation
Evaluate SE Practices and Environments	PARTIAL	Process Evaluation

ISO/IEC 12207 Element	Match	SEON View Concept
Evaluate Subcontractor's Software Products	PARTIAL	Process Evaluation
Evaluate Acquirer and Other Parties Required Support and Cooperation	PARTIAL	Process Evaluation
Evaluate Product and Process Measurements	PARTIAL	Process Evaluation
Evaluate Staff's Skill and Knowledge	PARTIAL	Process Evaluation
Quality Management Process	OVERLAP {*Organizational definitions not covered.*}	Quality Assurance Process
Quality Management Corrective Action	PARTIAL	Noncompliance Resolution
Take Corrective Actions	PARTIAL	Noncompliance Resolution
Implement Corrective Actions	OVERLAP {*Closing action not covered*}	Noncompliance Resolution
	OVERLAP {*Action resolution not covered*}	Noncompliance Closing
	PARTIAL	(Noncompliance Resolution + Noncompliance Closing)
Software Problem Resolution Process	OVERLAP {*Organizational definitions not covered*}	Quality Assurance Process
Problem Resolution	OVERLAP {*Problem Resolution deals with other problems besides NCs*}	Noncompliance Control
Report and Solve Problem	OVERLAP {*Report and Solve Problem deals with other problems besides NCs*}	Noncompliance Control
OBJECTS		
Quality Assurance Process Description	PARTIAL	Quality Assurance Plan
Quality Assurance Plan	EQUIVALENT	Quality Assurance Plan
Evaluation Record	EQUIVALENT	Evaluation Report
Nonconformance	EQUIVALENT	Noncompliance Register
Produced Artifact	ACTS AS	Evaluated Artifact
Corrective Action	EQUIVALENT	Corrective Action Register
Problem Resolutions	PARTIAL	Noncompliance Register
AGENTS		
Quality Evaluator	ACTS	Quality Auditor

Mapping Analysis: the ISO/IEC 12207 Software Quality Assurance and related Processes are covered in **90%** by the SEON View. The few points not covered are related to the communication of the process records and the establishment of organizational definitions.

A3.3. Develop the Integrated Content Model

Quality Assurance ICM built from a copy of the SEON View domain fragment (in blank) plus the new elements created for the ICM (in purple).



A total of **five new elements** were added to the Integrated Content Model (ICM): two activities for documenting and communicating the process results; two artifacts for register the process results and trends; and a team for perform such activities producing the artifacts. Elements added to the ICM and respective matches are shown in the table below.

Matches		New ICM Element	Definition
CMMI: Noncompliance Report	[P]	**Quality Assurance Control Report** (Document)	Document describing the quality assurance findings and results for monitoring and controlling purposes.
CMMI: Corrective Action Report	[P]		
CMMI: Evaluation Log	[P]		
CMMI: Quality Assurance Report	[P]		
CMMI: Corrective Action Status Report	[P]		
CMMI: Lesson Learned	[P]	**Quality Assurance Trends Report** (Document)	Document describing the main lessons and trends of the quality assurance activities.
CMMI: Quality Trends	[P]		
CMMI: Quality Trends Report	[P]		
CMMI: Quality Assurance Group	[E]	**Quality Assurance Group** (Stakeholder)	Team responsible for supporting the objective evaluation and noncompliance control activities, by documenting and reporting the quality assurance findings, results and trends.
CMMI: Provide Objective Insight	[W]	**Quality Results Documentation** (Composite Performed Activity)	Activity for documenting quality assurance information such as the achieved results, lessons learned, quality trends, and noncompliances, producing the QA Control and Trends Reports.
CMMI: Establish Records	[E]		
12207: Make Quality Assurance Records Available	[W]		
CMMI: Provide Objective Insight	[W]	**Quality Assurance Status Communication** (Composite Performed Activity)	Activity for communicating the quality assurance status and related information (evaluations, noncompliance resolution, corrective actions status, trends etc.) for the relevant stakeholders.
CMMI: Communicate and Resolve Noncompliance Issues	[O]		
CMMI: Identify and Involve Relevant Stakeholders	[O]		
12207: Make Quality Assurance Records Available	[O]		

Considering the ICM, the standards coverage achieved **95%** for CMMI-DEV SCM and **92%** for ISO/IEC 12207 SCM (against 61% and 91%, respectively, in the vertical mappings).

The remained uncovered elements, represented by nine partially covered elements (4 from CMMI and 5 from ISO 12207), are analyzed in the following table.

ELEMENT	ANALYSIS
CMMI-DEV	
Communicate and Resolve Noncompliance Issues [EVENT]	**Already Covered** This element can be analyzed considering two portions: Communication and Resolution of Noncompliance Issues. The Resolution part is covered in a [W] match with Noncompliance Resolution. The Communication part is covered in a [O] match with Quality Assurance Status Communication (ICM). Thus, it is considered covered.
Provide Objective Insight [EVENT]	**Already Covered** This element has exactly two parts: Establish Records (covered by Quality Results Documentation (ICM)) and Communicate and Resolve Noncompliance Issues (also covered). Thus, it is considered covered.
Identify and Involve Relevant Stakeholders [EVENT]	**Part Already Covered; Part Out of Scope** This element is a generic practice and has four matches: [O] Quality Assurance Planning: covering the identification and planning of stakeholders involvement; [O] Adherence Evaluation: covering the involvement in evaluations; [O] Noncompliance Control: covering the involvement in noncompliance control; [O] Quality Assurance Status Communication (from ICM): covering the involvement of interested stakeholders by communicating them. Thus, considering the defined initiative scope, the element is covered.
Institutionalize a Managed (PPQA) Process [EVENT]	**Part Already Covered; Part Out of Scope** Element introduced in the CMMI model only for organizing the generic practices. The uncovered part is out of scope.
ISO/IEC 12207	
Make Quality Assurance Records Available [EVENT]	**Already Covered** This element is covered by the sum of the new ICM Elements: [W] Quality Results Documentation and [W] Quality Assurance Status Communication.
Quality Management Process [EVENT]	**Part Already Covered; Part Out of Scope** The portion of this process relevant for the initiative is already covered. Other parts, related to organizational definitions are not in the harmonization scope.
Software Problem Resolution Process [EVENT]	**Part Already Covered; Part Out of Scope** The portion of this process relevant for the initiative is already covered. Other parts, related to organizational definitions are not in the harmonization scope.
Problem Resolution [EVENT]	**Part Already Covered; Part Out of Scope** The portion of this activity relevant for the initiative is already covered. Other parts, related to solving other types of problems are not in the harmonization scope.
Report and Solve Problem [EVENT]	**Part Already Covered; Part Out of Scope** The portion of this task relevant for the initiative is already covered. Other parts, related to solving other types of problems are not in the harmonization scope.

Considering this analysis, the CMMI-DEV and ISO/IEC 12207 SCMs for the Quality Assurance scope are fully covered (100%) by the Integrated Content Model.

E1. Extra Activity: Direct Mapping Standard to Standard

Mappings between the CMMI-DEV SCM and the ISO/IEC 12207 SCM. Matches with (D) denotes the ones Deduced from the previous mappings.

Standard to Standard Mapping: CMMI-DEV ⇨ ISO/IEC 12207 (Coverage: 70%)

CMMI-DEV Element	Match	ISO/IEC 12207 Element
EVENTS		
Process and Product Quality Assurance PA	WIDER (D)	Software Quality Assurance Process
	WIDER (D)	Process Implementation
	OVERLAP (D)	Quality Management Process
	OVERLAP (D)	Software Problem Resolution Process
Objectively Evaluate Processes and Work Products	WIDER (D) {*Objective evaluation not covered*}	Execute Quality Assurance Activities
	OVERLAP {*Only the communication of evaluations is covered*}	Make Quality Assurance Records Available
Objectively Evaluate Work Products	WIDER (D)	Product Assurance
	WIDER (D)	Evaluate Plans
	WIDER (D)	Evaluate Software Products and Related Documentation
	WIDER (D)	Evaluate Software Products for Delivery
Objectively Evaluate Processes	WIDER (D)	Evaluate Software Life Cycle Processes
	WIDER (D)	Evaluate SE Practices and Environments
	WIDER (D)	Evaluate Subcontractor's Software Products
	WIDER (D)	Evaluate Acquirer and Other Parties Required Support and Cooperation
	WIDER (D)	Evaluate Product and Process Measurements
	WIDER (D)	Evaluate Staff's Skill and Knowledge
	WIDER (D)	Process Assurance
Performed Process	EQUIVALENT (D)	Performed Process
Provide Objective Insight	PARTIAL {*Problem Resolution deals with other problems besides NCs*}	Problem Resolution
	PARTIAL {*Report and Solve Problem deals with other problems besides NCs*}	Report and Solve Problem
	OVERLAP {*Noncompliances not covered, only corrective actions resolution*}	Quality Management Corrective Action
Communicate and Resolve Noncompliance Issues	WIDER (D)	Quality Management Corrective Action
	WIDER (D)	Take Corrective Actions
	WIDER {*Noncompliance cause, analysis and communication not covered*}	Implement Corrective Actions
	OVERLAP {*Only communication of noncompliances is covered, not resolution*}	Make Quality Assurance Records Available
	PARTIAL	Problem Resolution
	PARTIAL	Report and Solve Problem

CMMI-DEV Element	Match	ISO/IEC 12207 Element
Establish Records	PARTIAL (D)	Make Quality Assurance Records Available
Institutionalize a Managed (PPQA) Process	NORELATION	
Plan the Process	PARTIAL (D)	Develop, Implement and Maintain Quality Assurance Plan
	WIDER {*Only the quality assurance process is established, not all the plan (evaluations, stakeholders etc.)*}	Establish Quality Assurance Process
Assign Responsibility	PARTIAL (D)	Develop, Implement and Maintain Quality Assurance Plan
Identify and Involve Relevant Stakeholders	OVERLAP (D) {*Involvement only planned not performed*}	Develop, Implement and Maintain Quality Assurance Plan
OBJECTS		
Evaluation Report	EQUIVALENT (D)	Evaluation Record
Noncompliance Report	PARTIAL {*Only report of noncompliances considered*}	Evaluation Record
Noncompliance Issue	EQUIVALENT (D)	Nonconformance
	WIDER (D)	Problem Resolutions
Lesson Learned	NORELATION	
Corrective Action	EQUIVALENT (D)	Corrective Action
Produced Work Product	EQUIVALENT (D)	Produced Artifact
Corrective Action Report	PARTIAL	Problem Resolutions
Quality Trends	NORELATION	
Evaluation Log	PARTIAL	Evaluation Record
Quality Assurance Report	PARTIAL	Evaluation Record
Corrective Action Status Report	PARTIAL	Problem Resolutions
Quality Trends Report	NORELATION	
Quality Assurance Plan	WIDER (D)	Quality Assurance Process Description
	EQUIVALENT (D)	Quality Assurance Plan
AGENTS		
Quality Assurance Group	NORELATION	
Project Staff	NORELATION	
Manager	NORELATION	

ISO/IEC 12207 SCM covers 70% of the CMMI SCM. The uncovered portion regards mostly objective evaluation restrictions and further details in the CMMI-DEV specification for stakeholders and work products.

Standard to Standard Mapping: ISO/IEC 12207 ⇨ CMMI-DEV (Coverage: 90%)

ISO/IEC 12207 Element	Match	CMMI-DEV Element
EVENTS		
Software Quality Assurance Process	PARTIAL (D) {*Needs the other processes (for managing nonconformities and corrective actions) to be complete*}	Process and Product Quality Assurance PA
Process Implementation	PARTIAL (D)	Process and Product Quality Assurance PA

ISO/IEC 12207 Element	Match	CMMI-DEV Element
Establish Quality Assurance Process	PARTIAL {*Only the quality assurance process is established, not all the plan (evaluations, stakeholders etc.)*}	Plan the Process
Develop, Implement and Maintain Quality Assurance Plan	WIDER (D)	Plan the Process
Develop, Implement and Maintain Quality Assurance Plan	WIDER (D)	Assign Responsibility
Develop, Implement and Maintain Quality Assurance Plan	OVERLAP (D)	Identify and Involve Relevant Stakeholders
Execute Quality Assurance Activities	PARTIAL (D) {*Execute Quality Assurance Activities does not objectively evaluate*}	Objectively Evaluate Processes and Work Products
Make Quality Assurance Records Available	WIDER (D)	Establish Records
Make Quality Assurance Records Available	OVERLAP {*Only the communication of evaluations is covered*}	Objectively Evaluate Processes and Work Products
Make Quality Assurance Records Available	OVERLAP {*Only communication of noncompliances is covered, not resolution*}	Communicate and Resolve Noncompliance Issues
Make Quality Assurance Records Available	PARTIAL	(Establish Records + Objectively Evaluate Processes and Work Products + Communicate and Resolve Noncompliance Issues)
Product Assurance	PARTIAL (D)	Objectively Evaluate Work Products
Evaluate Plans	PARTIAL (D)	Objectively Evaluate Work Products
Evaluate Software Products and Related Documentation	PARTIAL (D)	Objectively Evaluate Work Products
Evaluate Software Products for Delivery	PARTIAL (D)	Objectively Evaluate Work Products
Performed Process	EQUIVALENT (D)	Performed Process
Process Assurance	PARTIAL (D)	Objectively Evaluate Processes
Evaluate Software Life Cycle Processes	PARTIAL (D)	Objectively Evaluate Processes
Evaluate SE Practices and Environments	PARTIAL (D)	Objectively Evaluate Processes
Evaluate Subcontractor's Software Products	PARTIAL (D)	Objectively Evaluate Processes
Evaluate Acquirer and Other Parties Required Support and Cooperation	PARTIAL (D)	Objectively Evaluate Processes
Evaluate Product and Process Measurements	PARTIAL (D)	Objectively Evaluate Processes
Evaluate Staff's Skill and Knowledge	PARTIAL (D)	Objectively Evaluate Processes
Quality Management Process	OVERLAP (D) {*Organizational definitions not covered.*}	Process and Product Quality Assurance PA
Quality Management Corrective Action	PARTIAL (D)	Communicate and Resolve Noncompliance Issues
Quality Management Corrective Action	OVERLAP {*Noncompliances not covered, only corrective actions resolution*}	Provide Objective Insight
Take Corrective Actions	PARTIAL (D)	Communicate and Resolve Noncompliance Issues
Implement Corrective Actions	PARTIAL {*Noncompliance cause, analysis and communication not covered*}	Communicate and Resolve Noncompliance Issues
Software Problem Resolution Process	OVERLAP (D) {*Organizational definitions not covered*}	Process and Product Quality Assurance PA
Problem Resolution	WIDER {*Problem Resolution deals with other problems besides NCs*}	Provide Objective Insight
Problem Resolution	WIDER	Communicate and Resolve Noncompliance Issues

ISO/IEC 12207 Element	Match	CMMI-DEV Element
Report and Solve Problem	WIDER {*Report and Solve Problem deals with other problems besides NCs*}	Provide Objective Insight
	WIDER	Communicate and Resolve Noncompliance Issues
OBJECTS		
Quality Assurance Process Description	PARTIAL (D)	Quality Assurance Plan
Quality Assurance Plan	EQUIVALENT (D)	Quality Assurance Plan
Evaluation Record	EQUIVALENT (D)	Evaluation Report
	WIDER {*Only report of noncompliances considered*}	Noncompliance Report
	WIDER	Evaluation Log
	WIDER	Quality Assurance Report
Nonconformance	EQUIVALENT (D)	Noncompliance Issue
Produced Artifact	EQUIVALENT (D)	Produced Work Product
Corrective Action	EQUIVALENT (D)	Corrective Action
Problem Resolutions	PARTIAL (D)	Noncompliance Issue
	WIDER	Corrective Action Report
	WIDER	Corrective Action Status Report
AGENTS		
Quality Evaluator	NORELATION	

CMMI-DEV SCM covers 91% of the ISO/IEC 12207 SCM. The uncovered portion regards ISO elements that are only partially in the scope (such as Problem Resolution, that is partially covered by Noncompliance control, but the other types of problems solved are not in the scope of this initiative). Considering the Quality Assurance initiative scope, no relevant portion remained uncovered.

Final Considerations

Considering the defined scope for this initiative, regarding Quality Assurance processes, the vertical mappings supported the identification of matches between the modeled standards portions and the SEON View. In this phase, the SEON view has covered 61% of the CMMI-DEV defined scope and 90% of the ISO/IEC 12207 defined scope. The uncovered parts were resolved by adding new elements to the Integrated Content Model. Five elements remained not totally covered and were justified.

During the direct mappings between the standards, several matches were deduced from the vertical ones (marked with D) and reviewed. New matches were established between the considered standards' models achieving a coverage of 70% of CMMI-DEV SCM in relation to ISO/IEC 12207 SCM, and 90% in the inverse direction.

Appendix B
Questionnaire and Interviews Scripts

Questions used for the questionnaires and interviews applied in the Empirical Studies 2 and 3.

I. Questionnaire Script

Questionnaire for registering feedback about the Standards Harmonization Study.

1. *Identification*

- Name
- E-mail
- Academic Degree
- Academic / Professional Area

2. *Experience*

2.1 Experience with Quality Standards

- Experience Degree: [none, low (< 1 year), medium (1-3 years), high (> 3 years)]
- How have you acquired this experience? [course, project, professionally, other]

2.2 Experience with Conceptual Modeling / Ontologies

- Experience Degree: [none, low (< 1 year), medium (1-3 years), high (> 3 years)]
- How have you acquired this experience? [course, project, professionally, other]

2.3 Experience with the subject Domain

- Experience Degree: [none, low (< 1 year), medium (1-3 years), high (> 3 years)]
- How have you acquired this experience? [course, project, professionally, other]

3. *Difficulties for Performing the Activities*

3.1 Difficulties with Quality Standards

- Difficulty degree for understanding and extracting information from the standards: [very easy, easy, medium, high, very high]
- Describe the main difficulties.

3.2 Difficulties with Ontologies

- Difficulty degree for understanding, selecting and applying the SEON ontologies: [very easy, easy, medium, high, very high]
- Describe the main difficulties.

3.3 Difficulties with Conceptual Models

- Difficulty degree for creating the conceptual models: [very easy, easy, medium, high, very high]
- Describe the main difficulties.

3.4 Difficulties with the Harmonization Approach

- Difficulty degree for understanding and applying Harmony: [very easy, easy, medium, high, very high]
- Describe the main difficulties.

3.5 Difficulties with the Mapping Tool

- Difficulty degree for understating and using Mapper: [very easy, easy, medium, high, very high]
- Describe the main difficulties.

3.6 Difficulties with the Approach Activities

- Inform the difficulty degree for performing each Harmony activity: [very easy, easy, medium, high, very high]
  - A.1.1. Identify the Harmonization Purpose
  - A1.2. Define the Harmonization Scope
  - A1.3. Select the Standards' Portions
  - A1.4. Select the SEON View
  - A2.1. Develop the Standards' Structural Models
  - A2.2. Map Standards' Structural Models to Core Ontologies
  - A2.3. Develop the Integrated Structural Model
  - A3.1. Develop the Standards' Content Models
  - A3.2. Map Standards' Content Models to Domain Ontologies
  - A3.3. Develop the Integrated Content Model
  - E1. Extra Activity: Direct Mapping Standard to Standard (if required)
- Describe the main informed difficulties and how they were solved.

3.7 Information Available

- Was the available information (in the approach, tool, by the authors) sufficient for performing the activities?
- What else could contribute?

4. Activities Supports

4.1 Ontology's Support

- How much have the SEON ontologies contributed to the harmonization initiative? [1-5]
- Describe the main ontologies' contributions.

4.2 Approach's Support

- How much have Harmony contributed to the harmonization initiative? [1-5]
  - Phases and activities (step-by-step process) provided by the approach.
  - Scope as basis for selecting the standards and ontologies portions.
  - Structural models as basis for creating the content models.
  - Structural mappings as basis for the content mapping (basetype restrictions).
  - Ontologies as reference for the mappings and integrated models (ISM and ICM).
  - Match types provided.
  - Matches deductions in standard to standard mappings (if required).
  - The approach in general.
- Describe the main approaches' contributions.

4.3 Tool's Support

- How much have Mapper contributed to the harmonization initiative? [1-5]
- Describe the main tools' contributions.

5. Final Questions

5.1 Standards Inconsistencies

- During the activities have you identified any inconsistence in the involved standards?
- Which ones?

5.2 Improvement Suggestions

- Do you have any suggestion for Harmony?
- Do you have any suggestion for Mapper?

5.3 Complementary Information

- Do you have any complementary comment (about the standards, models / ontologies, approach, tool, obtained results, or the study)?

II. Interview Script

Script for conducting the interviews.

1. *Adopted Process and Practices*

- Describe the process, as you performed it, for the harmonization initiative.
- Have you applied any additional technique or practice for any activity (e.g., standards study, models creation, auxiliary schemes or tables, text highlighting)?

2. *Previous Experience*

- Do you have any previous knowledge or experience that helped performing the initiative? (e.g., knowledge / experience on the domain, ontologies, standards, software process, mapping / integration, other)

3. *Difficulties*

- Which were the main difficulties faced along the initiative?
- How were they solved?

4. *Inconsistencies*

- Have you identified any inconsistency in the standards? (e.g., structure, vocabulary, meanings)
- How have you dealt with them?

5. *Initiative Support*

- What could better support the initiative efforts? (e.g., additional information, other guidelines, better explanation, other tool features)

6. *Approach Benefits*

- Have you perceived any benefits by using Harmony?
- Can you list the main ones? (e.g., approach guidance, provided techniques, model-orientation, ontologies use, tool features, quality of results)
- Would you apply Harmony again in a similar situation?

7. *Final Considerations*

- Do you have any additional comment or suggestion?

Annex A

Harmonizing Software Testing Processes

This annex presents the report of an initiative harmonizing the Software Testing processes of ISO/IEC 12207, ISO/IEC 29110 and ISO/IEC 29119. This initiative occurred in an empirical study of Harmony conducted by a Software Testing expert. The results are organized according to the Harmony activities.

A.1. Purpose and Scope Definition

A.1.1. Identify the Harmonization Purpose

Provide a Harmonized Model for the Software Testing Processes of ISO/IEC 12207, ISO/IEC 29110 and ISO/IEC 29119.

A1.2. Define the Harmonization Scope

Dynamic Testing Processes, excluding aspects related to Test Management (such as test planning and monitoring) and Test Environment (e.g., test environment set-up and maintenance), as well as test organizational aspects (such as organizational test polices and strategies).

A1.3. Select the Standards' Portions

ISO/IEC 12207: Software Construction Process (7.1.5); Software Integration Process (7.1.6); Software Qualification Testing Process (7.1.7); and Software Validation Process (7.2.5).

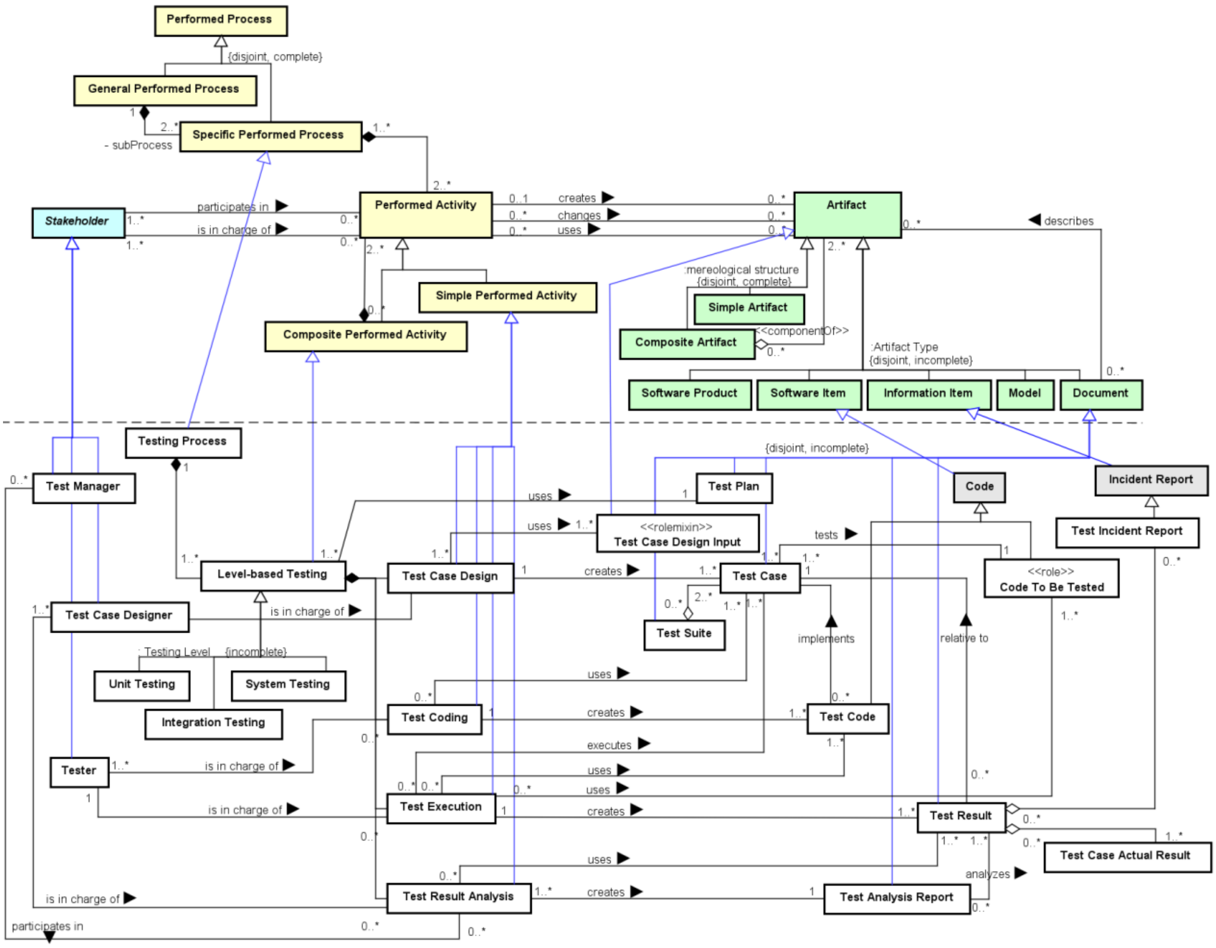
ISO/IEC 29110: Software Implementation (SI) process (7), testing activities and tasks.

ISO/IEC 29119: 29119-1: Software Testing Concepts (5.1), The Test Process (5.2.1); 29119-2: Dynamic Test Processes (8), except Test Environment Set-Up & Maintenance Process (8.3); and 29119-3: Dynamic Test Processes Documentation (7).

A1.4. Select the SEON View

SPO (core ontology): portion dealing with processes, activities, stakeholders and artifacts.

ROoST (domain ontology): Testing Process sub-ontology, as shown in the diagram below.

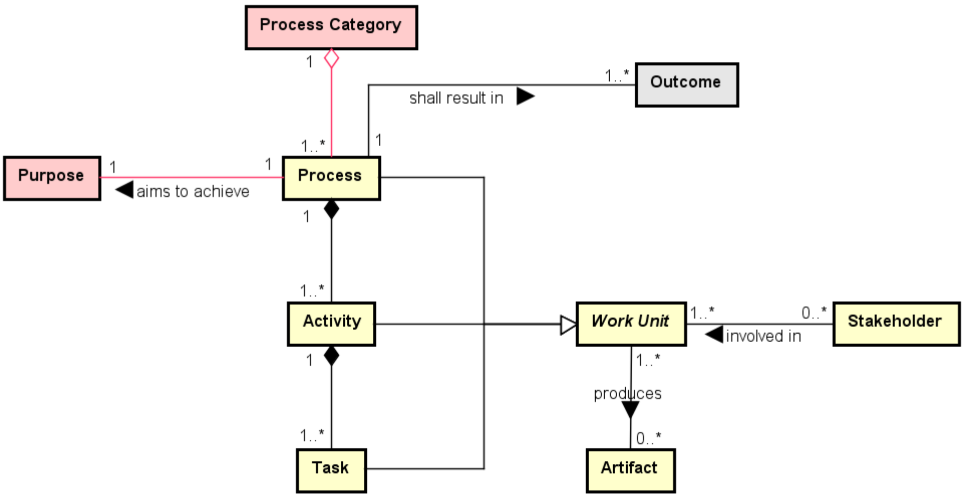


A.2. Structural Harmonization

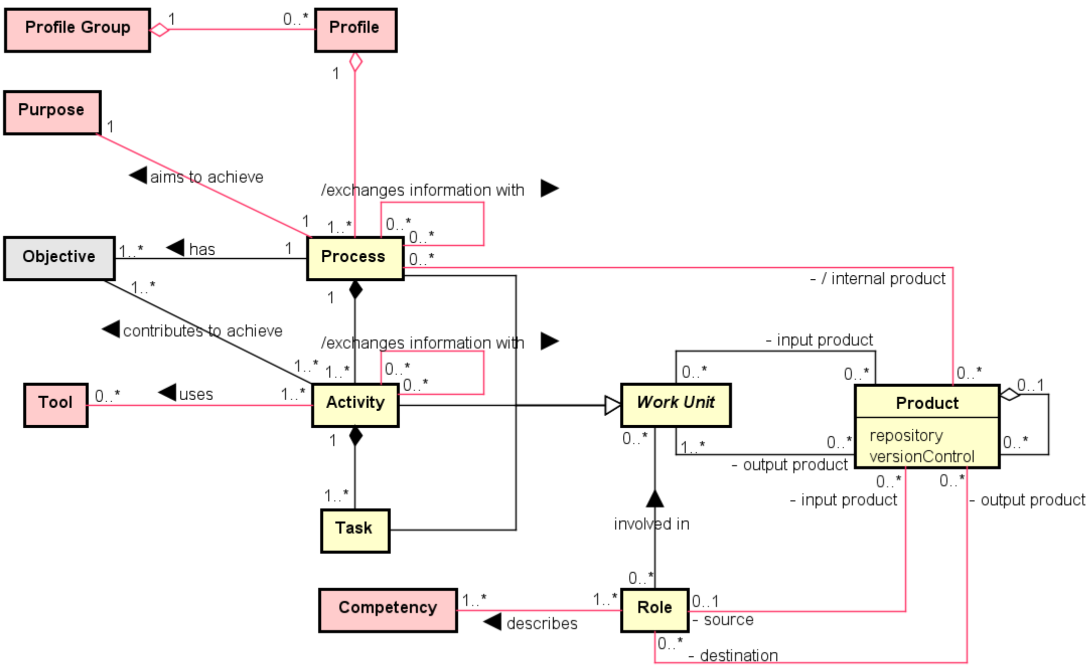
A2.1. Develop the Standards' Structural Models

Three Standards' Structural Models (SSMs) were developed. Elements in yellow and relations in black are those considered for being extended in the initiative (according to the defined scope); elements and relations in pink were discarded; elements in gray are used only as information source.

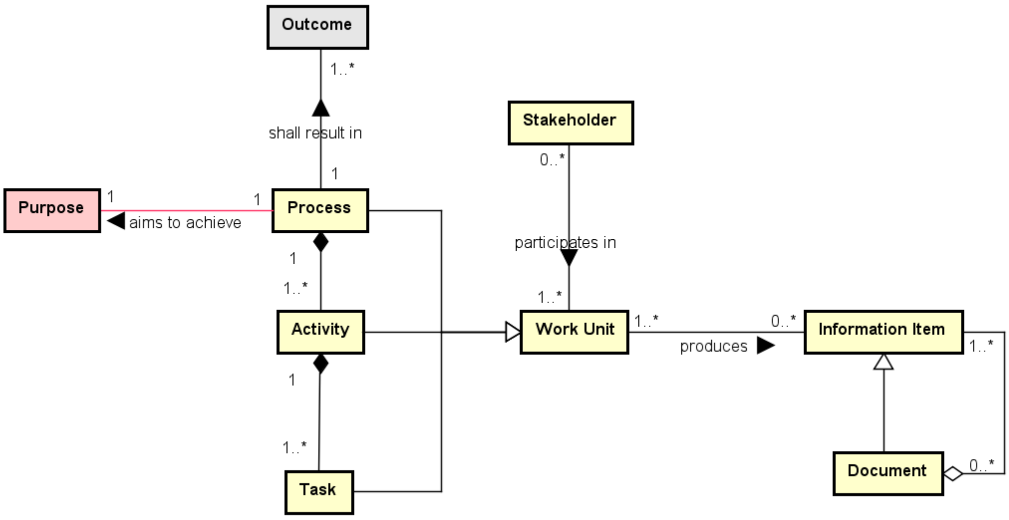
ISO/IEC 12207 SSM



ISO/IEC 29110 SSM



ISO/IEC 29119 SSM



A2.2. Map Standards' Structural Models to Core Ontologies

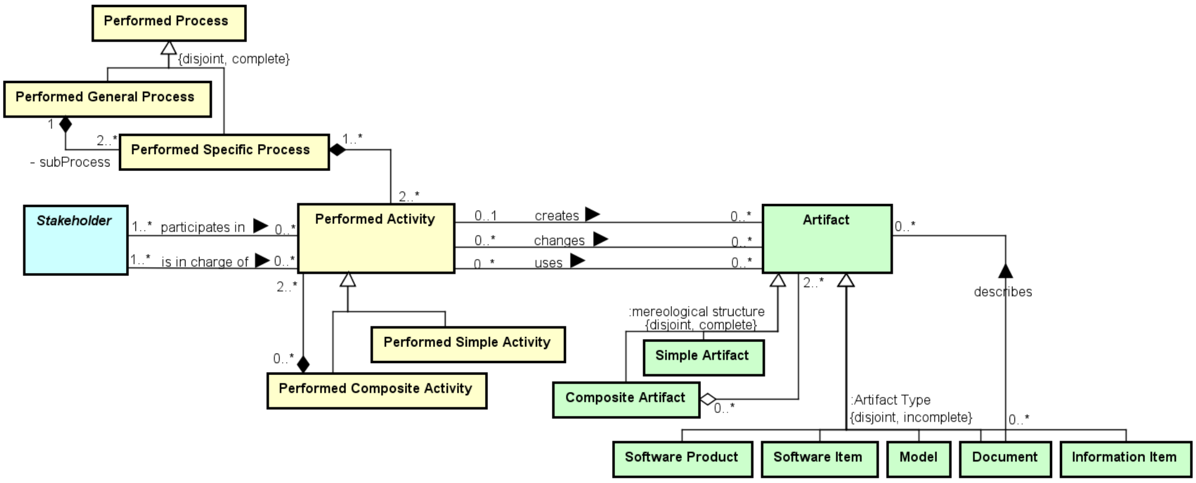
The Core Ontologies' concepts and standards elements are classified according to the ontological types (*event*, *object*, *agent* and *moment*). The standards' elements considered are compared to the core concepts with the same ontological type and matches are established between them (vertical mappings), as shown in the table.

Core Ontology Concept	ISO 29110 Element	ISO 12207 Element	ISO 29119 Element
Performed Process (event)	Process (event) Activity (event)	Process (event) Activity (event)	Process (event)
Performed Activity (event)	Activity (event) Task (event)*	Process (event) Activity (event) Task (event)*	Process (event) Activity (event) Task (event)*
Artifact (object)	Product (object)	Artifact (object)	Information Item (object) Document (object)
Stakeholder (agent)	Role (agent)	Stakeholder (agent)	Stakeholder (agent)

* Task is an element considered only when it denotes an <u>event</u>. Otherwise it will be used as an element for information.

A2.3. Develop the Integrated Structural Model

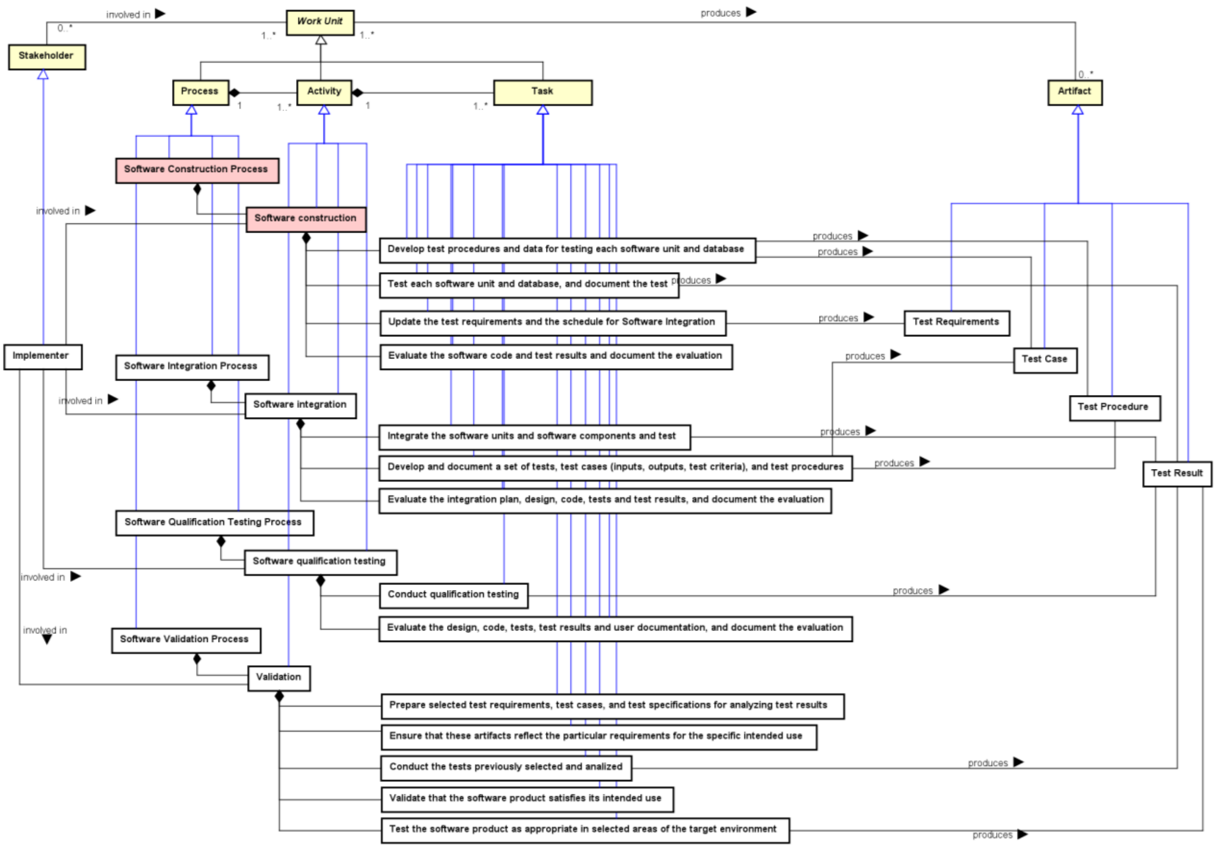
All the SSM elements were covered by the core concepts of the SEON View. Thus, there are no modifications to be made in the ISM and it corresponds to the core portion of the SEON View, as shown below.



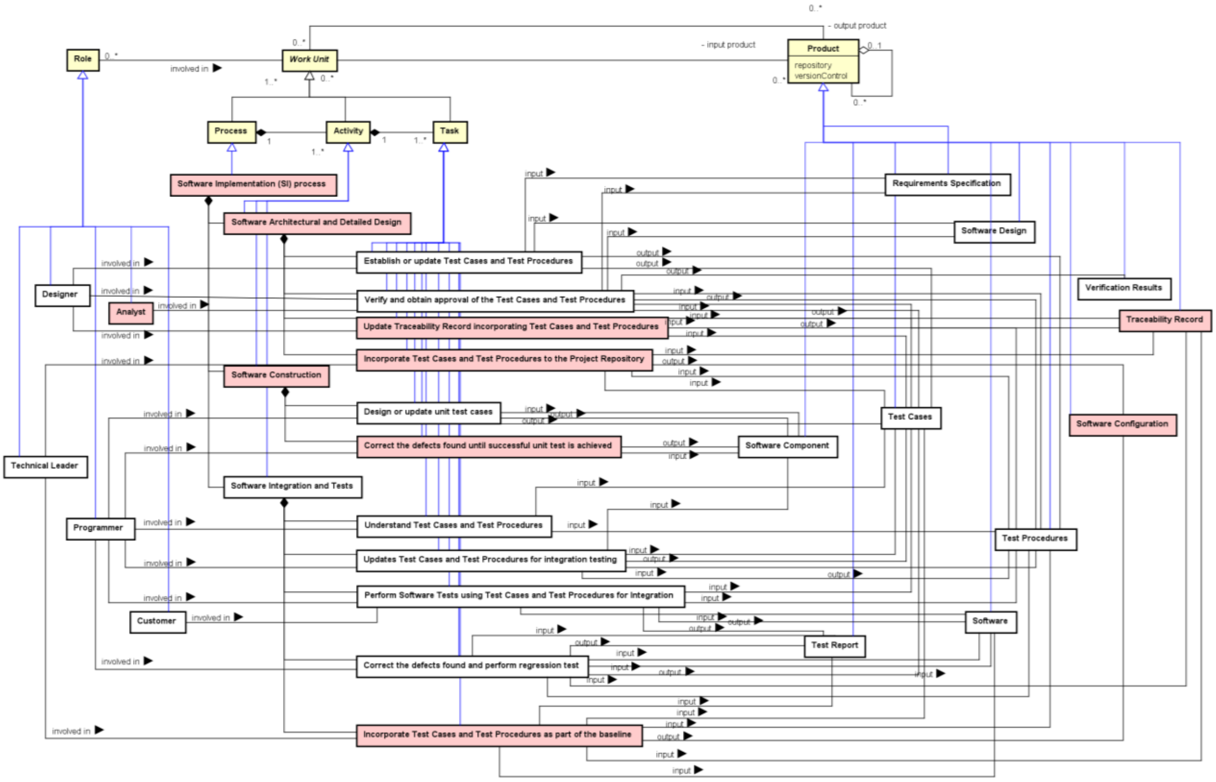
A.3. Content Harmonization

A3.1. Develop the Standards' Content Models

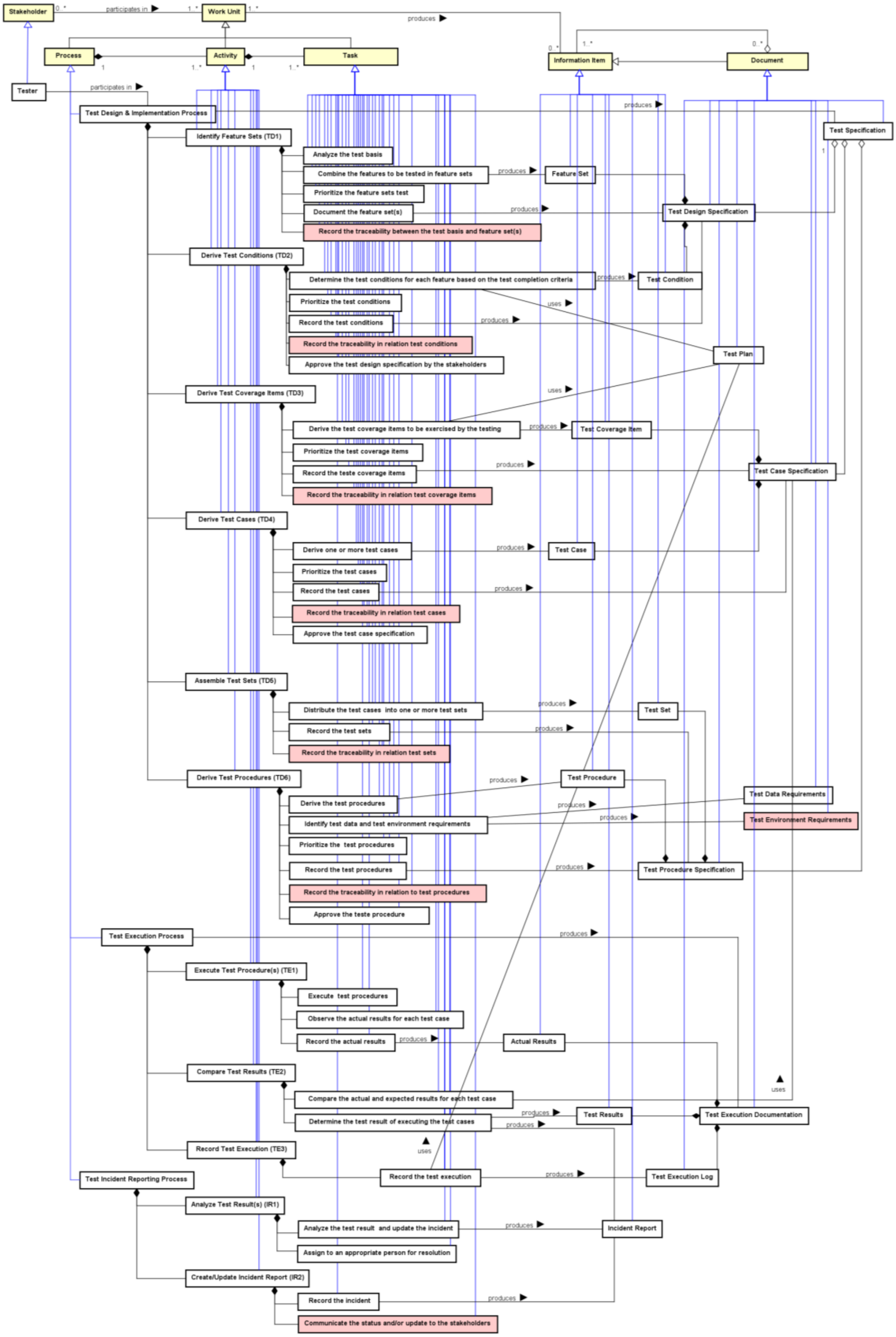
ISO/IEC 12207 SCM: created by specializing the ISO/IEC 12207 SSM considering the defined standard portion.



ISO/IEC 29110 SCM: created by specializing the ISO/IEC 29110 SSM considering the defined standard portion.



ISO/IEC 29119 SCM: created by specializing the ISO/IEC 29119 SSM considering the defined standard portion.



A3.2. Map Standards' Content Models to Domain Ontologies

Background colors in the first column points out the elements coverage: green for fully covered; yellow for partially covered; blank for not covered. Composite matches are shown with the target (third column) including two or more concepts.

Content Vertical Mapping: ISO/IEC 12207 ⇨ SEON View (Coverage: 63%[15])

ISO/IEC 12207 Element	Match	SEON View Concept
EVENTS		
Develop test procedures and data for testing each software unit and database	OVERLAP {*Procedimentos de teste não são tratados em Test Case Design. Assim, aqui há Overlap*}	Test Case Design
Test each software unit and database, and document the test	PARTIAL {*Refere-se à execução dos testes de unidades de software. Dessa forma, esse elemento foi considerado como sendo parte de Test Execution que é mais abrangente*}	Test Execution
Update the test requirements and the schedule for Software Integration	NORELATION	
Evaluate the software code and test results and document the evaluation	PARTIAL {*Refere-se à análise dos resultados da execução dos testes de unidades de software. Dessa forma, esse elemento foi considerado como sendo parte de Test Result Analysis que é mais abrangente*}	Test Result Analysis
Software Integration Process	OVERLAP {*Software Integration Process inclui a integração em si e depois o teste*}	Integration Testing
Software integration	OVERLAP {*Software Integration inclui a integração em si e depois o teste*}	Integration Testing
Integrate the software units and software components and test	EQUIVALENT {*Refere-se à integração de unidade de software e posteriormente os testes dessa integração.*}	Test Execution
Develop and document a set of tests, test cases (inputs, outputs, test criteria), and test procedures	OVERLAP {*Overlap, já que test procedures não são tratados em Test Case Design*}	Test Case Design
Evaluate the integration plan, design, code, tests and test results, and document the evaluation	OVERLAP {*Refere-se à análise dos resultados dos testes de integração. Existe um Overlap, já que há avaliação de coisas que não são relativas a teste (tal como plano de integração e design)*}	Test Result Analysis
Software Qualification Testing Process	PARTIAL	Testing Process
Software qualification testing	PARTIAL	Testing Process

[15] **Coverage** is an approximated number, calculated with a basic algorithm, considering the coverage done by the matches established for each standard element. An element coverage can be 0% (no matches), 33% (partially covered), 67% (largely covered), 0.67% to 90% (multiple partial matches) and 100% (fully covered). The mapping coverage ranges from 0% to 100% and is given by the sum of each individual element coverage divided by the number of elements (Σec/Σe). It does not take into account complexity or types of the elements, just the set coverage and quantity.

ISO/IEC 12207 Element	Match	SEON View Concept
Conduct qualification testing	PARTIAL {*Refere-se a execução dos testes de qualificação. Além disso, os resultados do teste devem ser registrados. Dessa forma, esse elemento foi considerado como sendo equivalente a execução de teste (Atividade de Test Execution na ontologia)*}	Test Execution
Evaluate the design, code, tests, test results and user documentation, and document the evaluation	OVERLAP {*Refere-se à análise dos resultados dos testes de qualificação. Existe um Overlap, já que há avaliação de coisas que não são relativas a teste (tal como documentação de usuário, código e design). Além disso, inclui avaliar Feasibility of operation and maintenance, que está fora da alçada de Test Result Analysis.*}	Test Result Analysis
Software Validation Process	OVERLAP {*É uma intersecção, uma vez que estamos tratando apenas de teste dinâmicos*}	Testing Process
Validation	OVERLAP {*É uma intersecção, uma vez que estamos tratando apenas de teste dinâmicos*}	Testing Process
Prepare selected test requirements, test cases, and test specifications for analyzing test results	NORELATION	
Ensure that these artifacts reflect the particular requirements for the specific intended use	NORELATION	
Conduct the tests previously selected and analized	PARTIAL	Test Execution
Validate that the software product satisfies its intended use	PARTIAL	Test Result Analysis
Test the software product as appropriate in selected areas of the target environment	PARTIAL	Test Execution
OBJECTS		
Test Requirements	NORELATION	
Test Case	EQUIVALENT	Test Case
Test Procedure	NORELATION	
Test Result	EQUIVALENT	Test Result
AGENTS		
Implementer	ACTS {*Como tudo é feito pelo "Implementer", então Implementer Acts as todos os demais papéis*}	Test Manager
	ACTS {*Como tudo é feito pelo "Implementer", então Implementer Acts as todos os demais papéis*}	Test Case Designer
	ACTS {*Como tudo é feito pelo "Implementer", então Implementer Acts as todos os demais papéis*}	Tester

Mapping Analysis: the ISO/IEC 12207 model is covered in 63% by the SEON View. The uncovered portion regards the update the test requirements, schedule for software integration (considered outside the initiative scope), activities and documentation of test procedures, and test conditions.

Content Vertical Mapping: ISO/IEC 29110 ⇨ SEON View (Coverage: 63%)

ISO/IEC 29110 Element	Match	SEON View Concept
EVENTS		
Establish or update Test Cases and Test Procedures	PARTIAL {*Estabelecer e atualizar casos de teste são parte de Test Case Design, mas estabelecer ou atualizar procedimentos de teste não. ROoST não trata de procedimentos de teste e, portanto, foi considerado como overlap*}	Test Case Design
Verify and obtain approval of the Test Cases and Test Procedures	OVERLAP {*ROoST não trata de procedimentos de teste e, portanto, aqui há overlap.*}	Test Case Design
Design or update unit test cases	PARTIAL	Test Case Design
Software Integration and Tests	OVERLAP {*A atividade de teste da norma compõe diversas tarefas que correspondem às atividades de teste do processo de teste, porém focado apenas em testes de integração. Além disso, é integrado o software e depois testado. Dessa forma, foi considerado como sendo overlap da atividade Integration Testing*}	Integration Testing
Understand Test Cases and Test Procedures	PARTIAL {Apesar de envolver a compreensão de Test Procedures, algo não tratado por ROoST, foi considerado parte de Test Execution, pois refere-se apenas à compreensão daquilo que será testado}	Test Execution
Updates Test Cases and Test Procedures for integration testing	OVERLAP {*ROoST não trata procedimentos de teste. Assim, atualizar casos de teste e procedimentos de teste é considerado como orverlap de Test Case Design*}	Test Case Design
Perform Software Tests using Test Cases and Test Procedures for Integration	PARTIAL {Apesar de envolver a execução de Test Procedures, algo não tratado por ROoST, foi considerado parte de Test Execution, pois refere-se apenas à execução de testes}	Test Execution
Correct the defects found and perform regression test	OVERLAP {*Esta atividade envolve a re-execução de casos de teste (teste de regressão) e, portanto, há um overlap com Test Execution*}	Test Execution
OBJECTS		
Requirements Specification	ACTS {*Requeriments Specification Acts as (desempenha o papel de) Test Case Design Input*}	Test Case Design Input
Software Design	ACTS {*Software Design Acts as (desempenha o papel de) Test Case Design Input*}	Test Case Design Input
Verification Results	PARTIAL	Test Result
Test Cases	EQUIVALENT	Test Case
Software Component	ACTS {*Software Component desempenha o papel de Code to Be Tested*}	Code To Be Tested
Test Procedures	NORELATION	
Software	ACTS {*Software pode desempenhar o papel de (Act as) Code to Be Tested*}	Code To Be Tested

ISO/IEC 29110 Element	Match	SEON View Concept
Test Report	WIDER {*Os resultados de vários resultados de teste são documentados em um Test Report. Assim, Test Report é definido como Wider than Test Result*}	Test Result
AGENTS		
Designer	ACTS {*Designer pode desempenhar o papel de Test Case Designer em atividades de elaboração de casos de testes*}	Test Case Designer
Technical Leader	ACTS {*Technical Leader desempenha o papel de Test Manager sobretudo em atividades de teste de unidade*}	Test Manager
Programmer	ACTS {*Programmer desempenha o papel de (Acts as) Test Case Designer sobretudo em atividades de teste de unidade e de integração*}	Test Case Designer
	ACTS {*Programmer desempenha o papel de (Acts as) tester sobretudo em atividades de teste de unidade*}	Tester
Customer	ACTS {*Customer desempenha o papel de (Acts as) Tester quando executa testes de sistema.* }	Tester

Mapping Analysis: ISO/IEC 29110 model is covered in 63% by the SEON View. The uncovered portion regards the activities and documentation of Test Procedures and Correction the Defects Found, the later is considered outside the initiative scope.

Content Vertical Mapping: ISO/IEC 29119 ⇨ SEON View (Coverage: 45%)

ISO/IEC 29119 Element	Match	SEON View Concept
EVENTS		
Test Design & Implementation Process	WIDER {*Test Design & Implementation tem muito mais coisas que Test Case Design. Assim, foi considerado como Wider*}	Test Case Design
Identify Feature Sets (TD1)	NORELATION	
Analyze the test basis	NORELATION	
Combine the features to be tested in feature sets	NORELATION	
Prioritize the feature sets test	NORELATION	
Document the feature set(s)	NORELATION	
Derive Test Conditions (TD2)	NORELATION	
Determine the test conditions for each feature based on the test completion criteria	NORELATION	
Prioritize the test conditions	NORELATION	
Record the test conditions	NORELATION	
Approve the test design specification by the stakeholders	NORELATION	
Derive Test Coverage Items (TD3)	NORELATION	

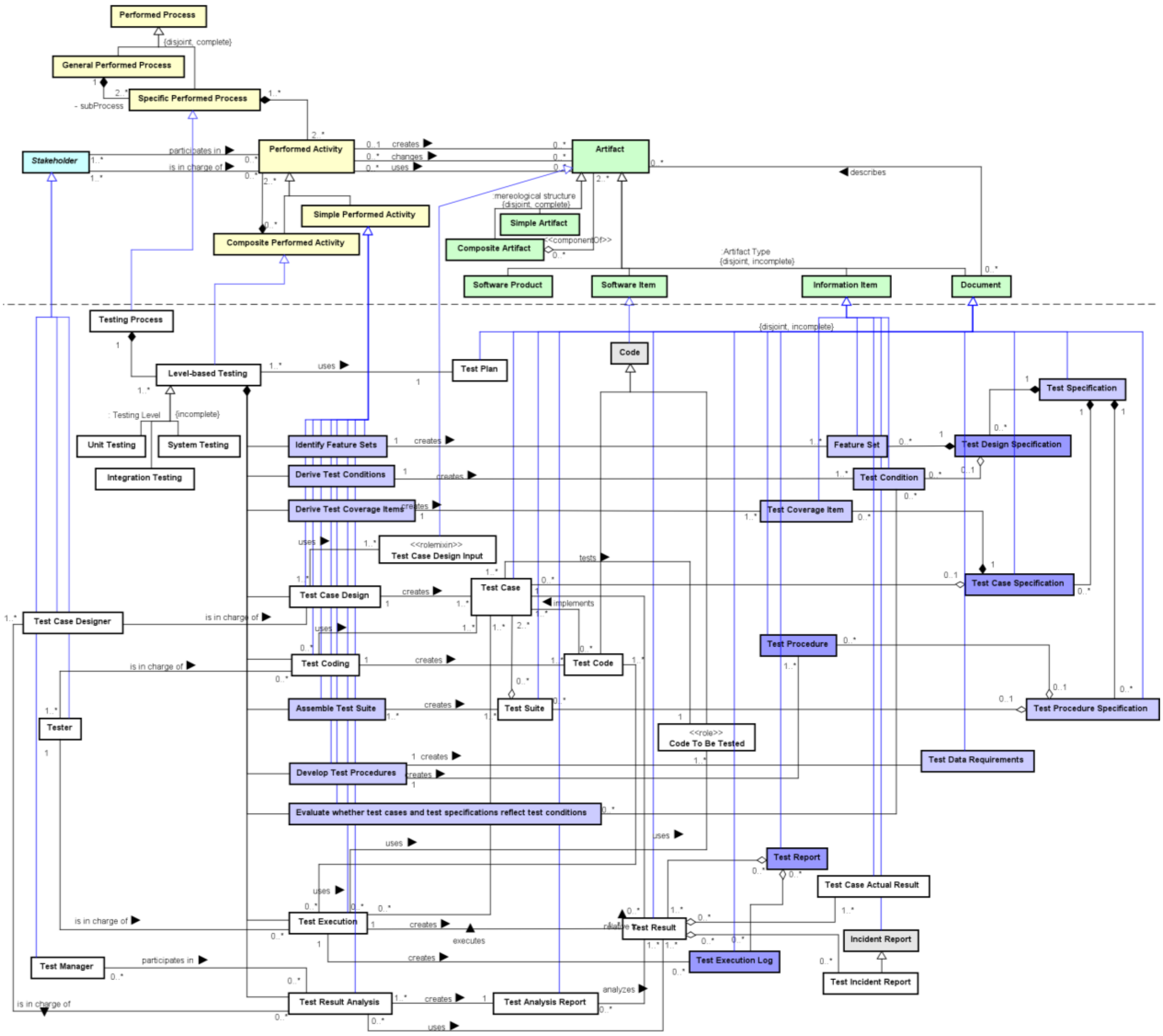
ISO/IEC 29119 Element	Match	SEON View Concept
Derive the test coverage items to be exercised by the testing	NORELATION	
Prioritize the test coverage items	NORELATION	
Record the test coverage items	NORELATION	
Derive Test Cases (TD4)	EQUIVALENT	Test Case Design
Derive one or more test cases	PARTIAL	Test Case Design
Prioritize the test cases	PARTIAL	Test Case Design
Record the test cases	PARTIAL	Test Case Design
Approve the test case specification	NORELATION	
Assemble Test Sets (TD5)	NORELATION	
Distribute the test cases into one or more test sets	NORELATION	
Record the test sets	NORELATION	
Derive Test Procedures (TD6)	NORELATION	
Derive the test procedures	NORELATION	
Identify test data and test environment requirements	NORELATION	
Prioritize the test procedures	NORELATION	
Record the test procedures	NORELATION	
Approve the test procedure	NORELATION	
Test Execution Process	EQUIVALENT	Test Execution
Execute Test Procedure(s) (TE1)	PARTIAL	Test Execution
Execute test procedures	PARTIAL	Test Execution
Observe the actual results for each test case	PARTIAL	Test Execution
Record the actual results	PARTIAL	Test Execution
Compare Test Results (TE2)	EQUIVALENT {*No momento em que o teste vai sendo executado é comparado o resultado atual com o esperado para determinar se o teste passou ou falhou e esse resultado do teste é cadastrado no documento de test results*}	Test Execution
Compare the actual and expected results for each test case	PARTIAL	Test Execution
Determine the test result of executing the test cases	PARTIAL	Test Execution
Record Test Execution (TE3)	PARTIAL	Test Execution
Record the test execution	PARTIAL	Test Execution
Test Incident Reporting Process	OVERLAP {*Test Incident Reporting é um processo que trata da análise de resultados e da criação (e algum gerenciamento) de incidentes*}	Test Result Analysis
Analyze Test Result(s) (IR1)	PARTIAL	Test Result Analysis
Analyze the test result and update the incident	PARTIAL	Test Result Analysis
Assign to an appropriate person for resolution	PARTIAL	Test Result Analysis

ISO/IEC 29119 Element	Match	SEON View Concept
Create / Update Incident Report (IR2)	OVERLAP {*Update Incident Report não é contemplada por Test Execution, apenas a primeira (Record the incident). Dessa forma, é considerado um overlap de Test Execution*}	Test Execution
Record the incident	PARTIAL	Test Execution
OBJECTS		
Test Specification	NORELATION	
Feature Set	NORELATION	
Test Design Specification	NORELATION	
Test Condition	NORELATION	
Test Plan	EQUIVALENT {*De acordo com a norma, Test Plan é detailed description of test objectives to be achieved and the means and schedule for achieving them, organized to coordinate testing activities for some test item or set of test items. Mesma definição para Test Plan na ontologia*}	Test Plan
Test Coverage Item	NORELATION	
Test Case Specification	WIDER {*Este documento tem mais informações do que o Test Case. Test Case é uma seção dele (7.3.5). Assim, ele é considerado Wider*}	Test Case
Test Case	EQUIVALENT	Test Case
Test Set	EQUIVALENT	Test Suite
Test Procedure	NORELATION	
Test Data Requirements	NORELATION	
Test Procedure Specification	WIDER {*Este documento contém tanto Test Procedures quanto Test Suites. Assim, ele é considerado Wider*}	Test Suite
Actual Results	EQUIVALENT	Test Case Actual Result
Test Results	PARTIAL	Test Result
Test Execution Documentation	WIDER {*Este documento contém Test Results, Actual Results e Test Execution Log. Assim, ele é considerado Wider.*}	Test Result
	WIDER {*Este documento contém Test Results, Actual Results e Test Execution Log. Assim, ele é considerado Wider.*}	Test Case Actual Result
Test Execution Log	NORELATION	
Incident Report	EQUIVALENT	Test Incident Report
AGENTS		
Tester	EQUIVALENT	Tester

Mapping Analysis: the ISO/IEC 29119 model is covered in 45% by the SEON View. Several elements were not covered by this standard. It is believed that this is due to the fact that the standard presents a great level of detail in relation to the processes, activities and tasks. Some uncovered portion regards are the Feature Sets, Test Conditions, Test Coverage Items, Assemble Test Sets and Test Procedures.

A3.3. Develop the Integrated Content Model

Software Testing ICM built from a copy of the SEON View domain fragment (in blank) plus the new elements created for the ICM (in purple).



A total of **16 new elements** were added to the Integrated Content Model (ICM), being six activities, seven documents, and four information items. Elements added to the ICM and respective matches are shown in the table below.

Matches		New ICM Element	Definition
12207: Develop test procedures and data for testing each software unit and database	[O]	**Develop Test Procedures** (Simple Performed Activity)	A atividade de procedimentos de teste tem como objetivo de ordenar os passos para execução dos casos de teste de acordo com as dependências descritas por pré-condições e pós-condições e outros requisitos de teste.
12207: Develop and document a set of tests, test cases (inputs, outputs, test criteria), and test procedures	[O]		
29110: Verify and obtain approval of the Test Cases and Test Procedures	[O]		
29110: Updates Test Cases and Test Procedures for integration testing	[O]		
29119: Derive Test Procedures (TD6)	[E]		
29119: Derive the test procedures	[P]		
29119: Identify test data and test environment requirements	[P]		
29119: Prioritize the test procedures	[P]		
29119: Record the test procedures	[P]		
29119: Approve the teste procedure	[P]		
29119: Test Data Requirements	[E]	**Test Data Requirements** (Document)	Nesse documento é definido os dados de teste, a origem dos dados do teste e o estado em que os dados de teste específicos estão localizados, dentre outros.
29119: Identify Feature Sets (TD1)	[E]	**Identify Feature Sets** (Simple Performed Activity)	Essa atividade tem como objetivo analisar as características / recursos de teste, ou seja, é realizado um estudo das características do que será testado (requisitos) e da divisão em conjuntos de testes.
29119: Analyze the test basis	[P]		
29119: Combine the features to be tested in feature sets	[P]		
29119: Prioritize the feature sets test	[P]		
29119: Document the feature set(s)	[P]		
12207: Update the test requirements and the schedule for Software Integration	[O]	**Derive Test Conditions** (Simple Performed Activity)	Condições de teste para cada recurso ou característica de teste devem ser determinados. Uma condição de teste é um aspecto testável de um componente ou sistema, como uma função, transação, recurso, atributo de qualidade ou elemento estrutural identificado como base para o teste. A condição de teste pode ser aplicada utilizando técnicas ou critérios de teste, por exemplo, classe de equivalência.
29119: Derive Test Conditions (TD2)	[E]		
29119: Determine the test conditions for each feature based on the test completion criteria	[P]		
29119: Prioritize the test conditions	[P]		
29119: Record the test conditions	[P]		
29119: Approve the test design specification by the stakeholders	[P]		
29119: Derive Test Coverage Items (TD3)	[E]	**Derive Test Coverage Items** (Simple Performed Activity)	Tem como objetivo medir a cobertura dos testes definidos nas condições de teste
29119: Derive the test coverage items to be exercised by the testing	[P]		
29119: Prioritize the test coverage items	[P]		
29119: Record the teste coverage items	[P]		
29119: Approve the test case specification	[P]		
29119: Assemble Test Sets (TD5)	[E]	**Assemble Test Suite** (Simple Performed Activity)	Essa atividade tem como objetivo montar os conjuntos de teste. Os casos de teste podem ser distribuídos em um ou mais conjuntos de teste com base em restrições em sua execução.
29119: Distribute the test cases into one or more test sets	[P]		
29119: Record the test sets	[P]		

Matches		New ICM Element	Definition
12207: Prepare selected test requirements, test cases, and test specifications for analyzing test results	[P]	**Evaluate whether test cases and test specifications reflect test requirements (conditions)** (Simple Performed Activity)	A atividade tem como objetivo preparar os requerimentos de teste, casos de teste e especificação de teste.
12207: Ensure that these artifacts reflect the particular requirements for the specific intended use	[P]		
29119: Test Coverage Item	[E]	**Test Coverage Item** (Information Item)	Item de informação gerado pela atividade Derive Test Coverage Items. Contém os itens de cobertura definidos nas condições de teste.
29119: Test Specification	[E]	**Test Specification** (Document)	Documentação completa do projeto de teste, casos de teste e procedimentos de teste
29119: Feature Set	[E]	**Feature Set** (Information Item)	Identify Feature Sets tem como produto (item de informação) gerado um Feature Set.
12207: Test Procedure	[E]	**Test Procedure** (Information Item)	Em Test Procedure são apresentadas as ações necessárias para configurar as pré-condições iniciais dos casos de testes e quaisquer atividades de encerramento de execução posterior
29110: Test Procedures	[E]		
29119: Test Procedure	[E]		
12207: Test Requirements	[E]	**Test Condition** (Information Item)	Derive Test Conditions tem como produto o o item de informação Test Condition.
29119: Test Condition	[E]		
29119: Test Case Specification	[E]	**Test Case Specification** (Document)	Documenta os conjuntos de casos de teste
29119: Test Case Specification	[E]	**Test Case Specification** (Document)	Documenta os conjuntos de casos de teste
29119: Test Design Specification	[E]	**Test Design Specification** (Document)	O Documento Test Design Specification apresenta os recursos a serem testados e suas condições de teste correspondentes
29110: Test Report	[E]	**Test Report** (Document)	Test Report pode ser composto por vários Test Results e Test Execution log
29119: Test Execution Documentation	[E]		
29119: Test Execution Log	[E]		
29119: Test Procedure Specification	[E]	**Test Procedure Specification** (Document)	No Documento de Test Procedure Specification são descritas as ações associadas que podem ser necessárias para configurar os casos de testes

These new ICM elements were matched with the remaining uncovered elements of ISO/IEC 12207, ISO/IEC 29110 and ISO/IEC 29119, increasing the coverage, respectively, from 63%, 63% and 45% to 82%, 73% and 98%.

The remained uncovered elements are analyzed in the following table.

ELEMENT	ANALYSIS
12207	
Develop test procedures and data for testing each software unit and database [EVENT]	**Already Covered** Desenvolver procedimento de teste já foi contemplado no ICM (Develop Test Procedures) em uma relação de overlap e a geração de dados para teste foram considerados no mapeamento com overlap de Test Case Design.
Software Integration Process [EVENT]	**Already Covered** Software Integration Process inclui a integração em si e depois o teste. Esse Elemento da norma foca apenas em testes de integração, sendo considerado orverlap com Integration Testing. Logo, é considerado com já coberto.

ELEMENT	ANALYSIS
Software integration [EVENT]	**Already Covered** Software Integration inclui a integração em si e depois o teste. Esse Elemento da norma foca apenas em testes de integração, sendo considerado orverlap com Integration Testing. Logo, é considerado com já coberto.
Develop and document a set of tests, test cases (inputs, outputs, test criteria), and test procedures [EVENT]	**Already Covered** Desenvolver procedimento de teste já foi contemplado no ICM (Develop Test Procedures) em uma relação de overlap e o desenvolvimento e documentação de casos de teste já foram considerados no mapeamento com overlap de Test Case Design.
Evaluate the integration plan, design, code, tests and test results, and document the evaluation [EVENT]	**Already Covered** Existe avaliação de coisas que não são relativas a teste (tal como plano de integração e design). A avaliação de teste e resultados de teste já foram considerados no mapeamento, logo o escopo já foi contemplado.
Evaluate the design, code, tests, test results and user documentation, and document the evaluation [EVENT]	**Already Covered** Existe avaliação de coisas que não são relativas a teste. A avaliação de teste e resultados de teste já foram considerados no mapeamento, logo o escopo já foi contemplado.
Software Validation Process [EVENT]	**Already Covered** Uma vez que o escopo dessa iniciativa é tratar apenas de teste dinâmicos, considera-se esse elemento como coberto.
Validation [EVENT]	**Already Covered** Uma vez que o escopo dessa iniciativa é tratar apenas de teste dinâmicos, considera-se esse elemento como coberto.
Implementer [AGENT]	**Already Covered** Na norma 12207 tudo é feito pelo “Implementer”, então Implementer age em todos os demais papéis. As atuações do implementer relacionadas ao escopo da iniciativa estão cobertas.
Update the test requirements and the schedule for Software Integration [EVENT]	**Part Already Covered; Part Out of Scope** Requisitos de teste foram contemplados nessa iniciativa pelo ICM. No entanto, "schedule for Software Integration" não faz parte do escopo dessa inciativa.
29110	
Verify and obtain approval of the Test Cases and Test Procedures [EVENT]	**Already Covered** Desenvolver procedimento de teste já foi contemplado no ICM (Develop Test Procedures) em uma relação de overlap e a aprovação de casos de teste foi considerada no mapeamento com overlap de Test Case Design.
Updates Test Cases and Test Procedures for integration testing [EVENT]	**Already Covered** Atualizar procedimento de teste já foi contemplado no ICM (Develop Test Procedures) em uma relação de overlap e a atualização de casos de teste foi considerada no mapeamento com overlap de Test Case Design.
Correct the defects found and perform regression test [EVENT]	**Part Already Covered; Part Out of Scope** Esta atividade envolve a re-execução de casos de teste (teste de regressão) e, portanto, há um overlap com Test Execution. A correção dos defeitos não faz parte do escopo dessa iniciativa.
Software Integration and Tests [EVENT]	**Part Already Covered; Part Out of Scope** A atividade de teste da norma compõe diversas tarefas que correspondem às atividades de teste do processo de teste, porém focado apenas em testes de integração. Além disso, é integrado o software e depois testado. Dessa forma, foi considerado como sendo overlap da atividade Integration Testing. A Integração de Software não faz parte do escopo de harmonização.
Requirements Specification [OBJECT]	**Already Covered** Requirements Specification Acts as (desempenha o papel de) Test Case Design Input, dessa forma já foi contemplado nessa iniciativa.

ELEMENT	ANALYSIS
Software Design [OBJECT]	**Already Covered** Software Design Acts as (desempenha o papel de) Test Case Design Input, dessa forma já foi contemplado nessa iniciativa.
Software Component [OBJECT]	**Already Covered** Software Component desempenha o papel de Code to Be Tested. Dessa forma, foi contemplado nessa iniciativa.
Software [OBJECT]	**Already Covered** Software pode desempenhar o papel de (Act as) Code to Be Tested. Dessa forma, é considerado contemplado por essa inciativa.
Designer [AGENT]	**Already Covered** Designer pode desempenhar o papel de Test Case Designer em atividades de elaboração de casos de testes. Dessa forma, pode ser considerado contemplado por essa iniciativa.
Technical Leader [AGENT]	**Already Covered** Technical Leader desempenha o papel de Test Manager sobretudo em atividades de teste de unidade, logo pode ser considerado contemplado por essa iniciativa.
Programmer [AGENT]	**Already Covered** Programmer desempenha o papel de (Acts as) Test Case Designer e Tester sobretudo em atividades de teste de unidade e de integração, logo pode ser considerado contemplado por essa inciativa.
Customer [AGENT]	**Already Covered** Customer desempenha o papel de (Acts as) Tester quando executa testes de sistema, logo pode ser considerado contemplado nessa inciativa.
29119	
Create / Update Incident Report (IR2) [EVENT]	**Part Already Covered; Part Out of Scope** Update Incident Report não é contemplada no escopo da iniciativa, apenas a primeira (Record the incident). Dessa forma, o elemento é considerado como sendo já coberto.
Test Incident Reporting Process [EVENT]	**Part Already Covered; Part Out of Scope** Test Incident Reporting é um processo que trata da análise de resultados e da criação (e algum gerenciamento) de incidentes. No escopo dessa iniciativa é considerado apenas a análise de resultados, logo o escopo foi considerado coberto nessa iniciativa.
Test Design & Implementation Process [EVENT]	**Part Already Covered; Part Out of Scope** O elemento Test Design & Implementation Process é usado para derivar casos de teste e procedimentos de teste. Não há no ICM um elemento correspondente a este processo. Contudo, todas as suas atividades que estão no escopo da harmonização estão contempladas.

Considering this analysis, the three Standards' Models for the Dynamic Software Testing scope are **fully covered** by the **Integrated Content Model**.

Final Considerations

Considering the defined scope for this initiative, regarding Software Testing processes, the vertical mappings supported the identification of matches between the modeled standards portions and the SEON View. The not covered parts were resolved by adding new elements to the Integrated Content Model (which are registered as improvement suggestions for SEON).