

André Kapich Souza

Suporte automatizado para desenvolvimento de ontologias utilizando padrões ontológicos de domínio

Vitória, ES

2017

André Kapich Souza

Suporte automatizado para desenvolvimento de ontologias utilizando padrões ontológicos de domínio

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática.

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Programa de Pós-Graduação em Informática

Orientador: Prof^a. Dra. Renata Silva Souza Guizzardi

Vitória, ES

2017

Agradecimentos

Agradeço a Deus pela força, saúde, coragem e persistência. Muitas lutas e desafios foram travadas, mas até aqui nos tem ajudado o Senhor.

A minha querida esposa Gisele, um presente que Deus colocou em minha vida para me dar apoio e carinho e para me incentivar a lutar pelos objetivos. Também a ela minhas desculpas por, algumas vezes, não ter dispensado o tempo desejado.

Agradeço, também, a minha família pelo apoio, em especial minha mãe Alcina e meu irmão Cláudio, me incentivando sempre a não desistir diante as dificuldades encontradas. Pela compreensão das ausências em alguns momentos em virtude da dedicação aos estudos.

Aos meus amigos que sempre me apoiaram nesta caminhada. Caminhada esta que me concedeu o privilégio de conhecer novas pessoas que hoje fazem parte do meu grupo de amizade. A estes eu agradeço pela companhia e pela ajuda quando tive necessidade.

A professora Renata pela orientação e confiança depositada em meu trabalho, pela ajuda na construção do conhecimento, pelo incentivo e paciência.

Agradeço também aos membros da banca de defesa do mestrado, o professor Giancarlo e Maria Luiza, por reservarem um pouco de tempo participando da defesa e contribuindo com a melhoria deste trabalho.

Enfim, agradeço a todos os que estiveram de forma direta ou indireta envolvidos nesta caminha que hoje se encerra para que outras possam ser iniciadas.

Aqui fica o meu sincero, muito obrigado!

Resumo

A Engenharia de Ontologias tem evoluído bastante nas últimas décadas, com um número crescente de metodologias, ferramentas e aplicativos, que estão sendo propostos e experimentados na academia e na indústria. Por meio das ontologias, o conhecimento compartilhado de um domínio pode ser modelado para ser comunicado entre pessoas e sistemas automatizados. Com isso, a utilização de ontologias se torna uma importante ferramenta em diversas áreas do conhecimento para se estruturar, organizar e apoiar o compartilhamento dos conceitos que são inerentes a essas áreas. Além disso, com o uso de ontologias, a interoperabilidade entre sistemas se torna possível, devido à normatização e ao uso de padrões em sua construção. No entanto, o desenvolvimento de ontologias a partir do zero é uma tarefa difícil e complexa, uma vez que uma ontologia deve fornecer uma representação completa e coerente de uma parte específica do mundo. Assim, a reutilização é altamente recomendada em seu desenvolvimento, permitindo que as ontologias sejam construídas com base em modelos pré-existentes, levando a melhores resultados quanto a sua qualidade. Neste sentido, Padrões Ontológicos (OPs) são considerados como ferramentas interessantes para facilitar a reutilização. Recentemente, vários autores da comunidade de Engenharia de Ontologias já propuseram OPs e mecanismos para aplicá-los. No entanto, sistemas automatizados para apoiar a sua utilização na prática ainda são raros. Para preencher esta lacuna, esta dissertação propõe um editor para catálogos OPs, cujo objetivo é apoiar o gerenciamento e o reúso desses padrões. Assim, a abordagem de catálogo de OPs pode ser aplicada na construção de ontologias, com suporte automático. No desenvolvimento do editor proposto, optou-se por estender um editor de ontologias existente (o OLED) para aproveitar suas ferramentas de modelagem, verificação, transformação e validação. Também foram parcialmente implementados três catálogos de OPs específicos para os domínios de Serviço, de Processo de Software baseado na ISO e de Colaboração. Além disso, esta dissertação descreve três exemplos de utilização, um para cada um dos domínios citados, visando demonstrar a viabilidade da abordagem de construção de ontologias utilizando catálogos de OPs com o uso do editor desenvolvido, enfatizando também o benefício do reúso de OPs.

Palavras-chaves: Padrões Ontológicos, Suporte Automatizado, Catálogo de Padrões, Abordagem de Catálogos, Reúso de Padrões.

Abstract

Ontology Engineering has evolved a lot in the last decades, with a growing number of methodologies, tools and applications being proposed and tried out in academia and industry. Through ontologies, the shared knowledge of a domain can be modeled to be communicated between people and automated systems. With this, the use of ontologies becomes an important tool in several areas of knowledge to structure, organize and support the sharing of the concepts that are inherent to these areas. In addition, with the use of ontologies, interoperability between systems becomes possible, due to the standardization and the use of patterns in its construction. However, the development of ontologies from scratch is a difficult and complex task, since an ontology must provide a complete and coherent representation of a specific part of the world. Thus, reuse is highly recommended in its development, allowing the ontologies to be built based on pre-existing models, leading to better results as to their quality. In this sense, Ontological Patterns (OPs) are considered as interesting tools to facilitate reuse. Recently, several authors of the Ontology Engineering community have already proposed OPs and mechanisms to apply them. However, automated systems to support their use in practice are still rare. To fill this gap, this dissertation proposes an editor for OPs catalogues, whose objective is to support the management and reuse of these OPs. Thus, the OPs catalogue approach can be applied in the construction of ontologies, with automatic support. In the development of the proposed editor, we opted to extend an existing ontology editor (OLED) to take advantage of its modeling, verification, transformation and validation tools. Three specific OPs catalogues for the Service, ISO-based Software Process and Collaboration domains have also been partially implemented. In addition, this dissertation describes three examples of utilization, one for each of the mentioned domains, aiming to demonstrate the viability of the ontology building approach using OPs catalogues with the use of the developed editor, emphasizing also the benefit of the reuse of OPs.

Keywords: Ontology Patterns, Automated Support, Patterns Catalogue, Catalogues Approach, Patterns Reuse.

Lista de ilustrações

Figura 1 – Ilustração de uma ontologia segundo Swartout e Tate (GASEVIC; DJURIC; DEVEDZIC, 2009)	31
Figura 2 – Classificação de ontologias de acordo com a expressividade, propósito e especificidade (ISOTANI; BITTENCOURT, 2015)	31
Figura 3 – Distinção de tipos de objeto OntoUML (RUY et al., 2015)	33
Figura 4 – Fragmento do metamodelo da linguagem OntoUML para o tipo <i>Class</i> adaptado (GUIZZARDI, 2005)	34
Figura 5 – Fragmento do metamodelo da linguagem OntoUML para o tipo <i>Relationship</i> adaptado (GUIZZARDI, 2005)	34
Figura 6 – Tela de um projeto novo criado no OLED	37
Figura 7 – Suporte do OLED ao processo de desenvolvimento de ontologias (GUERSON et al., 2015)	38
Figura 8 – Tipos de padrões ontológicos (FALBO et al., 2013b)	42
Figura 9 – Níveis de generalidade e padrões ontológicos (RUY et al., 2015)	42
Figura 10 – Modelo do processo da S-OPL (FALBO et al., 2015a)	44
Figura 11 – Modelo conceitual do padrão <i>Organization Provider</i> (O-Provider) (FALBO et al., 2015a)	48
Figura 12 – Modelo conceitual do padrão <i>SOffering</i> (<i>Service Offering</i>) (FALBO et al., 2015a)	48
Figura 13 – Modelo conceitual do padrão <i>O-OU-Customer</i> (<i>Organization/Organizational Unit Customer</i>) (FALBO et al., 2015a)	49
Figura 14 – Modelo conceitual do padrão <i>DocD</i> (<i>Document Depiction</i>) (FALBO et al., 2015b)	51
Figura 15 – Modelo conceitual do padrão <i>PPas</i> (<i>Producer Participation simplified</i>) (FALBO et al., 2015b)	52
Figura 16 – Modelo conceitual do padrão <i>PPP</i> (<i>Project Process Performing</i>) (FALBO et al., 2015b)	53
Figura 17 – Modelo conceitual do padrão <i>PWUC</i> (<i>Performed Work Unit Composition</i>) (FALBO et al., 2015b)	54
Figura 18 – Modelo conceitual do padrão <i>PWUD</i> (<i>Performed Work Unit Dependence</i>) (FALBO et al., 2015b)	55
Figura 19 – Modelo conceitual do padrão <i>StD</i> (<i>Stakeholder Definition</i>) (FALBO et al., 2015b)	56
Figura 20 – Modelo conceitual do padrão <i>WUC</i> (<i>Work Unit Composition</i>) (FALBO et al., 2015b)	57

Figura 21 – Modelo conceitual do padrão WPChan (<i>Work Product Change</i>) (FALBO et al., 2015b)	58
Figura 22 – Modelo conceitual do padrão WPCrea (<i>Work Product Creation</i>) (FALBO et al., 2015b)	59
Figura 23 – Modelo conceitual do padrão WPUse (<i>Work Product Usage</i>) (FALBO et al., 2015b)	59
Figura 24 – Modelo conceitual do padrão WPN (<i>Work Product Nature</i>) (FALBO et al., 2015b)	60
Figura 25 – Ontologia de cooperação (OLIVEIRA, 2009a)	62
Figura 26 – As três subontologias que compõem a ontologia de colaboração (CONTO)	63
Figura 27 – Ontologia de comunicação (OLIVEIRA, 2009a)	64
Figura 28 – Fragmento da ontologia de coordenação (OLIVEIRA, 2009a)	65
Figura 29 – Fragmento da ontologia de coordenação representando conceitos de grupos e protocolos e as respectivas relações (OLIVEIRA, 2009a)	65
Figura 30 – Extração de OPs de um fragmento da ontologia de cooperação	67
Figura 31 – Collaborative Session Location	67
Figura 32 – Collaborative Session Action Contributions	68
Figura 33 – Collaborative Session Agents	69
Figura 34 – Collaborative Commitment for Action Contributions	69
Figura 35 – Collaborative Commitment Inheres in Collaborator	70
Figura 36 – Action Contribution Pre and Pos Situation	71
Figura 37 – Collaborative Session Resources Simple	71
Figura 38 – Resource Nature	72
Figura 39 – Processo da abordagem metodológica proposta	73
Figura 40 – Ontologia de referência de cooperação da CONTO com os padrões destacados	76
Figura 41 – Extensão do DROP RN adaptado (SOUZA et al., 2016)	77
Figura 42 – Modelo do processo da C-OPL	79
Figura 43 – Criação de um OP (<i>RN</i>) a partir de uma ontologia de referência	82
Figura 44 – Criação de um OP a partir do zero	84
Figura 45 – Criação de um diagrama (<i>RN</i>)	85
Figura 46 – OP <i>RN</i>	85
Figura 47 – Menu de acesso as configurações do catálogo de OPs	86
Figura 48 – Configuração do projeto OLED como catálogo de OPs	87
Figura 49 – Visualização do conteúdo do modelo do processo	88
Figura 50 – Configuração da intenção do catálogo de OPs	89
Figura 51 – Configuração da justificativa do catálogo de OPs	90
Figura 52 – Configuração do relacionamento entre os OPs do catálogo de OPs	92
Figura 53 – Configuração das CQs do OP	93

Figura 54 – Importação de um catálogo de OPs no OLED	95
Figura 55 – Escolha de um catálogo de OPs no OLED	96
Figura 56 – Visualização de um catálogo de OPs carregado no OLED	97
Figura 57 – Seleção do padrão CSR do catálogo	98
Figura 58 – Características do padrão CSR do catálogo	100
Figura 59 – Ajuda sobre as categorias OntoUML do OP	101
Figura 60 – Ávore de categorias de OntoUML	101
Figura 61 – Seleção do padrão RN do catálogo	103
Figura 62 – Características do padrão RN do catálogo	105
Figura 63 – Pré-visualização do OP RN	106
Figura 64 – Reúso da classe <i>Resource</i> do OP	106
Figura 65 – Ontologia parcial com os OPs <i>CSR</i> e <i>RN</i>	107
Figura 66 – Subontologia de oferta de serviço (QUIRINO et al., 2015)	111
Figura 67 – Acesso ao modelo do processo da S-OPL	112
Figura 68 – Visualização do modelo do processo da S-OPL	112
Figura 69 – Subontologia de oferta de serviço utilizando o OLED estendido	114
Figura 70 – Subontologia de acordo de serviço (FALBO et al., 2016)	115
Figura 71 – Processo de software baseado na ISP-OPL (RUY et al., 2015b)	117
Figura 72 – Ontologia do processo de requisitos (sub-ontologia do processo de definição de requisitos das partes interessadas) (RUY et al., 2015a)	118
Figura 73 – Acesso ao modelo do processo da ISP-OPL anexado ao catálogo de DROPs	119
Figura 74 – Visualização do processo da ISP-OPL anexado ao catálogo de DROPs	120
Figura 75 – Ontologia do processo de requisitos parcial (sub-ontologia do processo de definição de requisitos das partes interessadas)	121
Figura 76 – Criação do DROP (<i>RN</i>) a partir de uma ontologia de referência	123
Figura 77 – Alteração em lote de classe no OLED	124
Figura 78 – Ontologia para o editor colaborativo online com a utilização dos DROPs do catálogo de colaboração (SOUZA et al., 2016)	125
Figura 79 – Ontologia para o editor colaborativo online com a utilização dos DROPs do catálogo de colaboração	126
Figura 80 – Modelo do processo da C-OPL para o editor colaborativo	127
Figura 81 – Tempo de experiência em modelagem de ontologias	129
Figura 82 – Percepção da modelagem de ontologias	130
Figura 83 – Recursos utilizados em modelagem de ontologias	130
Figura 84 – Suporte utilizado em modelagem de ontologias	131
Figura 85 – Recursos presentes na ferramenta computacional para modelagem de ontologias	131
Figura 86 – Percepção do suporte computacional em modelagem de ontologias	132

Figura 87 – Avaliação do suporte automatizado e da abordagem metodológica para catálogo de DROPs	132
Figura 88 – Avaliação do suporte automatizado e da abordagem metodológica quanto a produtividade para catálogo de DROPs	133
Figura 89 – Avaliação do suporte automatizado e da abordagem metodológica quanto a qualidade dos catálogos de DROPs	134
Figura 90 – Avaliação do suporte automatizado e da abordagem metodológica quanto ao compartilhamento dos catálogos de DROPs	134
Figura 91 – Avaliação do suporte automatizado e da abordagem metodológica quanto a configuração dos catálogos de DROPs	135
Figura 92 – Avaliação do suporte automatizado na reutilização de catálogos de DROPs	136
Figura 93 – Avaliação do suporte automatizado para reutilização de catálogo de DROPs na modelagem de ontologias quanto à produtividade	136
Figura 94 – Avaliação do suporte automatizado para reutilização de catálogo de DROPs na modelagem de ontologias quanto à qualidade	137
Figura 95 – Avaliação do suporte automatizado para reutilização de catálogo de DROPs na modelagem de ontologias quanto ao compartilhamento	138
Figura 96 – Avaliação do suporte automatizado e da abordagem metodológica para catálogos de DROPs	138
Figura 97 – Contribuição da abordagem metodológica e do suporte automatizado para modelagem de ontologias	139
Figura 98 – Familiaridade com a abordagem de catálogos de DROPs com suporte automatizado	140
Figura 99 – Contribuição da abordagem metodológica e do suporte automatizado para catálogos de DROPs	140
Figura 100 – Contribuição da abordagem de catálogos de DROPs e o suporte automatizado para modelagem de ontologias	141

Lista de tabelas

Tabela 1 – Especificação parcial do padrão <i>Organization Provider</i> (O-Provider) (FALBO et al., 2015a) adaptado	47
Tabela 2 – Especificação parcial do padrão <i>SOffering</i> (<i>Service Offering</i>) (FALBO et al., 2015a) adaptado	48
Tabela 3 – Especificação parcial do padrão <i>O-OU-Customer</i> (<i>Organization/Organizational Unit Customer</i>) (FALBO et al., 2015a) adaptado	49
Tabela 4 – Especificação parcial do padrão <i>DocD</i> - <i>Document Depiction</i> (FALBO et al., 2015b) adaptado	51
Tabela 5 – Especificação parcial do padrão <i>PPas</i> - <i>Producer Participation simplified</i> (FALBO et al., 2015b) adaptado	52
Tabela 6 – Especificação parcial do padrão <i>PPP</i> - <i>Project Process Performing</i> (FALBO et al., 2015b) adaptado	53
Tabela 7 – Especificação parcial do padrão <i>PWUC</i> - <i>Performed Work Unit Composition</i> (FALBO et al., 2015b) adaptado	54
Tabela 8 – Especificação parcial do padrão <i>PWUD</i> - <i>Performed Work Unit Dependence</i> (FALBO et al., 2015b) adaptado	55
Tabela 9 – Especificação parcial do padrão <i>StD</i> - <i>Stakeholder Definition</i> (FALBO et al., 2015b) adaptado	56
Tabela 10 – Especificação parcial do padrão <i>WUC</i> - <i>Work Unit Composition</i> (FALBO et al., 2015b) adaptado	57
Tabela 11 – Especificação parcial do padrão <i>WPChan</i> - <i>Work Product Change</i> (FALBO et al., 2015b) adaptado	58
Tabela 12 – Especificação parcial do padrão <i>WPCrea</i> - <i>Work Product Creation</i> (FALBO et al., 2015b) adaptado	58
Tabela 13 – Especificação parcial do padrão <i>WPUse</i> - <i>Work Product Usage</i> (FALBO et al., 2015b) adaptado	59
Tabela 14 – Especificação parcial do padrão <i>WPN</i> - <i>Work Product Nature</i> (FALBO et al., 2015b) adaptado	60
Tabela 15 – Dicionário de termos da ontologia de cooperação (OLIVEIRA, 2009a) .	63
Tabela 16 – Dicionário de termos da ontologia de comunicação (OLIVEIRA, 2009a)	64
Tabela 17 – Dicionário de termos da ontologia de coordenação (OLIVEIRA, 2009a)	66
Tabela 18 – Especificação parcial do padrão <i>Collaborative Session Location</i> (CSL) .	67
Tabela 19 – Especificação parcial do padrão <i>Collaborative Session Action Contributions</i> (CSAC)	68
Tabela 20 – Especificação parcial do padrão <i>Collaborative Session Agents</i> (CSA) . .	68

Tabela 21 – Especificação parcial do padrão <i>Collaborative Commitment for Action Contributions</i> (CCAC)	69
Tabela 22 – Especificação parcial do padrão <i>Collaborative Commitment Inheres in Collaborator</i> (CCIC)	70
Tabela 23 – Especificação parcial do padrão <i>Action Contribution Pre and Pos Situation</i> (ACPSS)	70
Tabela 24 – Especificação parcial do padrão <i>Collaborative Session Resources Simple</i> (CSRS)	71
Tabela 25 – Especificação parcial do padrão <i>Resource Nature</i> (RN)	72
Tabela 26 – Especificação parcial do padrão <i>Resource Nature</i> (RN) (SOUZA et al., 2016)	77
Tabela 27 – Subontologia de oferta de serviço - mapeamento entre as questões de competência e os padrões	110
Tabela 28 – Mapeamento entre as questões de competência e os DROPs do catálogo de colaboração	125

Lista de abreviaturas e siglas

OLED	<i>OntoUML Lightweight Editor</i>
DROP	<i>Domain-related Ontology Pattern</i>
OP	<i>Ontology Pattern</i>
UFO	<i>Unified Foundational Ontology</i>
UML	<i>Unified Modelling Language</i>
CONTO	<i>Collaboration Ontology</i>
IA	Inteligência Artificial
SEMDM	Software Engineering Metamodel for Development Methodologies

Sumário

1	INTRODUÇÃO	21
1.1	Problema	23
1.2	Proposta	24
1.3	Metodologia	25
1.4	Validação	26
1.5	Estruturação	27
2	BASE TEÓRICA	29
2.1	Ontologia	29
2.2	OntoUML	33
2.3	OLED - OntoUML Lightweight Editor	36
2.4	Padrões Ontológicos	40
2.5	OPLs - Linguagens de Padrões Ontológicos	43
3	CATÁLOGO DE PADRÕES ONTOLÓGICOS DE DOMÍNIO	45
3.1	Domínio de Serviço	46
3.1.1	Padrões Ontológicos de Serviço	47
3.1.1.1	O-Provider - <i>Organization Provider</i>	47
3.1.1.2	SOffering - <i>Service Offering</i>	48
3.1.1.3	O-OU-Customer - <i>Organization/Organizational Unit Customer</i>	49
3.2	Domínio de Processo de Software	49
3.2.1	Padrões Ontológicos de Processo de Software	50
3.2.1.1	DocD - <i>Document Depiction</i>	51
3.2.1.2	PPas - <i>Producer Participation simplified</i>	52
3.2.1.3	PPP - <i>Project Process Performing</i>	52
3.2.1.4	PWUC - <i>Performed Work Unit Composition</i>	53
3.2.1.5	PWUD - <i>Performed Work Unit Dependence</i>	54
3.2.1.6	StD - <i>Stakeholder Definition</i>	55
3.2.1.7	WUC - <i>Work Unit Composition</i>	56
3.2.1.8	WPChan - <i>Work Product Change</i>	57
3.2.1.9	WPCrea - <i>Work Product Creation</i>	58
3.2.1.10	WPUse - <i>Work Product Usage</i>	59
3.2.1.11	WPN - <i>Work Product Nature</i>	59
3.3	Domínio de Colaboração	60
3.3.1	Ontologia de Colaboração (CONTO)	61
3.3.2	Padrões Ontológicos de Colaboração	66

3.3.2.1	Padrões de Cooperação	67
3.3.2.1.1	CSL – Collaborative Session Location	67
3.3.2.1.2	CSAC – Collaborative Session Action Contributions	68
3.3.2.1.3	CSA – Collaborative Session Agents	68
3.3.2.1.4	CCAC – Collaborative Commitment for Action Contributions	69
3.3.2.1.5	CCIC – Collaborative Commitment Inheres in Collaborator	69
3.3.2.1.6	ACPPS – Action Contribution Pre and Pos Situation	70
3.3.2.2	Padrões de Coordenação	71
3.3.2.2.1	CSRS – Collaborative Session Resources Simple	71
3.3.2.2.2	RN – Resource Nature	71
4	ABORDAGEM METODOLÓGICA PARA A CRIAÇÃO DE CATÁLOGOS DE DROPS	73
4.1	Selecionar a Ontologia de Referência	74
4.2	Selecionar o Módulo Ontológico	74
4.3	Extrair Padrões	75
4.4	Validar Padrões	78
4.5	OPL de Colaboração	78
5	SUPORTE AUTOMATIZADO PARA CATÁLOGOS DE PADRÕES ONTOLÓGICOS	81
5.1	Criação de Catálogos de Padrões Ontológicos	81
5.1.1	Criando um OP a partir de uma Ontologia de Referência	82
5.1.2	Criando um OP a partir do Zero	83
5.1.3	Configuração de um Catálogo de OPs	85
5.1.3.1	Intenção	89
5.1.3.2	Justificativa	90
5.1.3.3	Padrões Alternativos	91
5.1.3.4	Questões de Competência	92
5.2	Utilização de Catálogos de Padrões Ontológicos	94
5.2.1	Carregando um Catálogo de DROPs no OLED	94
5.2.2	Utilizando um Catálogo de DROPs no OLED	96
6	EXEMPLOS DE USO DE PADRÕES ONTOLÓGICOS PARA CONSTRUIR ONTOLOGIAS OPERACIONAIS	109
6.1	Domínio de Serviço	109
6.2	Domínio de Processo de Software	116
6.3	Domínio de Colaboração	122
6.4	Avaliação da Abordagem de Catálogos de DROPs	126
6.4.1	Planejamento da Avaliação	128

6.4.2	Resultados da Avaliação	128
6.4.2.1	Modelagem de Ontologias	129
6.4.2.2	Abordagem de Catálogo de DROPs	132
6.5	Considerações	141
7	TRABALHOS RELACIONADOS	143
7.1	NeOn Toolkit	143
7.2	WebODE	144
7.3	Protégé	145
7.4	Considerações Relevantes	146
8	CONCLUSÕES	147
8.1	Dificuldades	148
8.2	Contribuições	149
8.3	Trabalhos Futuros	149
	REFERÊNCIAS	151
	APÊNDICES	157
	APÊNDICE A – DOCUMENTO DE ESPECIFICAÇÃO DO CATÁ- LOGO DE DROPS DE COLABORAÇÃO	159
	APÊNDICE B – PESQUISA DE AVALIAÇÃO DO SUPORTE AU- TOMATIZADO E DA ABORDAGEM DE CATÁ- LOGO DE DROPS	189

1 Introdução

A área conhecida como Engenharia de Ontologias evoluiu bastante nas últimas décadas com um número crescente de metodologias, ferramentas e aplicações propostas e experimentadas na academia e na indústria. No entanto, o desenvolvimento de ontologias a partir do zero é ainda uma tarefa difícil e complexa, uma vez que se espera que uma ontologia forneça uma representação abrangente e coerente de uma porção específica do mundo (SOUZA et al., 2016). Em muitos casos, o engenheiro de ontologias não possui um conhecimento profundo do domínio em que a ontologia está sendo construída. Em outros casos, o especialista no domínio não possui o conhecimento técnico para construção de ontologias. Portanto, a construção de ontologias se torna demorada, exigindo que, tanto engenheiros de ontologias e quanto especialistas no domínio sejam altamente qualificados e, de preferência, tenham conhecimento aprofundado sobre as perspectivas uns dos outros do problema em questão. Como resultado, em geral, a qualidade das ontologias construídas fica comprometida.

Semelhante às abordagens bem-sucedidas e boas práticas de Engenharia de Software, a reutilização tem sido amplamente defendida para que as ontologias sejam construídas com base em modelos pré-existentes, acelerando o processo de desenvolvimento e levando a resultados de melhor qualidade.

Atualmente, aqueles que desejam reutilizar ontologias para a construção de novas ontologias, normalmente o fazem realizando buscas em repositórios de ontologias leves, que são aquelas que enfatizam a definição da taxonomia que representa a relação hierárquica entre conceitos e não se preocupam em definir detalhadamente cada conceito representado (ISOTANI; BITTENCOURT, 2015). De posse das ontologias selecionadas, é feita uma avaliação das mesmas em relação à tarefa ou domínio de interesse ou se utiliza ontologias de referência ou de núcleo que podem ser diretamente reutilizadas e especializadas (QUIRINO, 2016).

Neste contexto, os Padrões Ontológicos (OPs¹) têm sido apontados como ferramentas interessantes para permitir reutilização por se mostrarem uma abordagem promissora, favorecendo a reutilização de experiências codificadas e boas práticas. Os OPs são utilizados em soluções de modelagem para resolver problemas recorrentes de desenvolvimento de ontologias porque OPs, em geral, são veículos para encapsular o conhecimento. Eles são considerados um dos meios mais eficazes para nomear, organizar e raciocinar sobre modelagem do conhecimento (FALBO et al., 2013b).

¹ Apesar desta dissertação ser escrita em português, será utilizado o acrônimo OP porque é um acrônimo conhecido para padrões ontológicos em inglês. Logo o acrônimo em inglês será adotado.

Sendo assim, em muitas organizações, têm-se utilizado OPs como uma forma de modelagem que permite a troca de informações dentro e fora das suas fronteiras, fomentando a colaboração, por se acreditar que as organizações que colaboram com eficiência têm maior rendimento do que seu concorrente. Na verdade, a colaboração reduz os custos de transações entre as organizações. Com isso, as organizações podem criar modelos de negócios extremamente especializados que podem combinar um conjunto de serviços prestados pela colaboração entre organizações (FALBO et al., 2014).

No entanto, muitos desses modelos e padrões são criados de forma independente, sem necessariamente compartilhar a mesma semântica. Isso frequentemente dá origem a inconsistências entre eles e este problema é ainda mais ampliado quando padrões ou ontologias leves diferentes são usados em conjunto, causando problemas de interoperabilidade semântica. Alguns esforços foram feitos na criação de catálogos de OPs para suporte à criação de ontologias operacionais.

Diferentes tipos de OPs apoiam engenheiros de ontologias em fases distintas do processo de desenvolvimento de ontologias (FALBO et al., 2013a). Nesse contexto, ressaltam-se os catálogos de Padrões Ontológicos Relacionados a Domínio (DROPs²), que são definidos como um tipo de padrão ontológico de modelagem de conteúdo que captura um fragmento reutilizável extraído de uma ontologia de domínio de referência, apoiando a construção do modelo conceitual de uma nova ontologia de domínio, as ditas ontologias operacionais.

A reutilização de DROPs tem possibilitado acelerar o processo de desenvolvimento de ontologias, economizando tempo e dinheiro e promovendo a aplicação de boas práticas. Em particular, esforços têm sido feitos no desenvolvimento de redes de DROPs interconectados para fornecer suporte para resolver uma classe de problemas de desenvolvimento de ontologia para um domínio específico (FALBO et al., 2014). Essas redes são conhecidas como Linguagem de Padrões de Ontologia (OPLs) e buscam a melhoria do processo de construção de modelos mais consistentes, por meio da disseminação das melhores práticas de forma organizada e padronizada.

Embora os DROPs sejam úteis, aplicá-los na prática pode exigir muito esforço, uma vez que, em geral, os catálogos de DROPs são compostos por vários DROPs interligados (SOUZA et al., 2016). Uma boa solução para este problema é a aplicação de ferramentas computacionais que apoiem a criação e o uso de catálogos de DROPs na construção de ontologias. Análogo a outras áreas, também em engenharia de ontologias, um processo automatizado proporcionaria um ganho de produtividade e qualidade para a modelagem de ontologias baseada em DROPs. No momento, entretanto, essas ferramentas ainda são

² Apesar desta dissertação ser escrita em português, será utilizado o acrônimo DROP porque é um acrônimo conhecido para padrões ontológicos de relacionado ao domínio em inglês. Logo o acrônimo em inglês será adotado.

escassas.

Esta pesquisa visa cobrir a lacuna citada no parágrafo anterior. Para isso, propõe uma abordagem automatizada para criação e uso de catálogos de DROPs, desenvolvendo também uma ferramenta que permita ao engenheiro de ontologias construir um catálogo como esses, bem como utilizar os DROPs definidos nesses catálogos para a construção de ontologias.

1.1 Problema

Apesar do reúso ser uma boa prática reconhecida na área de Engenharia de Ontologias, a reutilização de ontologias é, em geral, um dos tópicos mais desafiadores e negligenciados dessa área. O problema de selecionar as melhores ontologias para reutilização, estendê-las e compor vários fragmentos de ontologia a partir delas ainda não foram adequadamente abordados (GANGEMI; PRESUTTI, 2009).

Em algumas situações, investir muito tempo e esforço na construção de uma ontologia ideal, tendo sempre um especialista do domínio e um especialista em modelagem para aprofundar a conceituação e a modelagem, pode valer à pena. Mas, em outros casos, isso não é desejável ou não é possível, pois isso torna custoso o processo de construção e manutenção de outras ontologias que as tomam como base (BLOMQVIST, 2009).

Uma outra situação que traz muitas dificuldades aos engenheiros de ontologias é a linguagem de representação utilizada na modelagem. Muitas vezes, essa linguagem não possui expressividade suficiente para representar todos os conceitos necessários para a modelagem, levando a modelagens ambíguas, incompletas e inconsistentes (VALASKI; REINEHR; MALUCELLI, 2016; ZHDANOVA; KELLER, 2005). Quando a linguagem utilizada na modelagem não consegue transmitir o conhecimento de forma precisa e clara, a qualidade do trabalho e o resultado ficam prejudicados. A identificação de uma linguagem, bem como suas características, é de fundamental importância porque a expressividade da linguagem tem grande influência na eficácia da comunicação do conhecimento do domínio assim como na praticidade de representá-lo (ZHDANOVA; KELLER, 2005; VALASKI; REINEHR; MALUCELLI, 2016).

Outros desafios relevantes da estruturação do conhecimento é o compartilhamento do conhecimento dos domínios (GRUBER, 1993). Diferentes comunidades têm suas conceituações compartilhadas inerentes aos seus integrantes. Desta forma, conceituações tendem a ter interpretações diferentes em comunidades distintas levando a inconsistências e ambiguidades no compartilhamento, o que é indesejável. Isso acaba se refletindo na modelagem, mesmo quando utilizamos a abordagem de catálogos de DROPs para a construção de ontologias, em que, muitas vezes, o engenheiro de ontologias não é especialista no domínio e nem dispõe de ferramenta adequada que lhe permita a construção de ontologias a partir

dos catálogos de DROPs.

1.2 Proposta

Esta dissertação se propõe a criar mecanismos para que os problemas citados na seção anterior sejam resolvidos ou, pelo menos, minimizados. Para isso, foi proposta a utilização da abordagem de catálogos de DROPs como uma forma de minimizar a necessidade da presença de um engenheiro de ontologias e de um especialista no domínio para a construção de ontologias operacionais nesses domínios. Isso se torna possível devido ao mapeamento dos problemas recorrentes de modelagem nas ontologias de referências desses domínios para OPs, o que transfere o conhecimento da ontologia de referência de domínio para o OP (FALBO, 2011). Com o catálogo de DROPs criado, basta agora, para os engenheiros de ontologias, a seleção de quais OPs do catálogo são de interesse para a modelagem da ontologia operacional desejada.

A construção de ontologias operacionais a partir de catálogos de DROPs não só deixa de exigir a presença de um especialista do domínio, como também proporciona que a modelagem da mesma não tenha necessidade de partir do zero. Isso também favorece o compartilhamento de conceituações e de soluções ontológicas baseadas nestas conceituações, tornando as ontologias em que estas soluções estão baseadas mais abrangentes, formais e bem fundamentas, por se utilizarem de todo o ferramental que elas oferecem. Outro ponto muito importante é a possibilidade da reutilização desses padrões aplicados na construção de ontologias uma vez que, conforme (FALBO et al., 2013a), padrões são veículos para encapsular conhecimento e são considerados um dos meios mais eficazes para a nomeação, organização e raciocínio sobre o conhecimento. Se esse compartilhamento for feito por meio de uma junção dos padrões que descrevem problemas recorrentes na forma de um catálogo, a produção de ontologias operacionais de domínio será ainda mais acelerada e há maiores possibilidades de que as ambiguidades e inconsistências dos modelos sejam removidas.

Embora OPs sejam úteis, aplicá-los, na prática, pode demandar muito esforço uma vez que, em geral, catálogos de DROPs são compostos de vários OPs isolados, que devem ser interligados na utilização. Com o apoio de uma ferramenta que ajude a interligar o OPs do catálogo na construção das ontologias operacionais, o processo de modelagem da ontologia se torna mais prático, ágil e sistemático. Assim, esta dissertação propõe a criação de um módulo para apoiar aos engenheiros de ontologias na construção de modelos a partir de catálogos de DROPs. Esse módulo é uma extensão do OLED, ferramenta desenvolvida para a construção de ontologias em OntoUML pelo Núcleo de Estudos em Modelagem Conceitual e Ontologias (NEMO), laboratório da Universidade Federal do Espírito Santo (UFES). Optou-se pela extensão do OLED para aproveitar todo o

ferramental de desenvolvimento e validação de ontologias desse editor. Assim, com a adição das novas funcionalidades ao OLED, ter-se-á o proveito das funcionalidades já existentes da ferramenta, visando o desenvolvimento de ontologias de mais alta qualidade.

1.3 Metodologia

Este trabalho foi desenvolvido seguindo os seguintes passos:

Aquisição de Conhecimento - Serviu para a aquisição de conhecimento necessário e amadurecimento das informações adquiridas. Isso se deu por meio de atividades que envolviam padrões, padrões ontológicos, linguagens de padrões, linguagens de padrões ontológicos, notações visuais, modelagem de ontologias, ferramentas de apoio a modelagem de ontologias e catálogo de padrões ontológicos por meio de artigos, teses, dissertações, aulas expositivas, trabalhos e estudos dirigidos.

Modelagem de Padrões Ontológicos - Tendo alcançado um nível satisfatório no passo anterior, se iniciaram as modelagens de OPs de colaboração a partir da ontologia de referência CONTO (OLIVEIRA, 2009a; OLIVEIRA, 2009b). Alguns DROPs de colaboração já haviam sido especificados e modelados, mas muitos outros ainda não tinham sido definidos. Logo, eles foram definidos e a modelagem dos DROPs passou a ser feita em uma ferramenta computacional de propósito geral. O objetivo era um catálogo de DROPs colaborativos. Com isso, também foi possível propor uma abordagem metodológica para a criação de catálogos de DROPs por meio da definição de passos que é apresentada no capítulo 4.

Suporte Automatizado para a Modelagem de Ontologias - Focou-se na criação de mecanismos para o gerenciamento de catálogos de DROPs como forma de utilizar a abordagem de catálogo de DROPs para a reutilização (SOUZA et al., 2016). Assim, foi verificado a possibilidade de extensão de alguma ferramenta já existente em detrimento do desenvolvimento de uma nova a partir do zero. Então o OLED foi escolhido como ferramenta, por já suportar a modelagem de ontologias das quais os DROPs e, conseqüentemente, os catálogos de DROPs muito se aproveitam. Outro ponto que justifica a escolha, é o fato do OLED implementar a linguagem OntoUML, que é uma linguagem formal e bem fundamentada, que muito favorece a modelagem pela familiaridade com a modelagem UML. Assim, um módulo foi criado no OLED para possibilitar o trabalho com catálogos de DROPs.

Criação de Catálogos de DROPs - Foram escolhidos alguns domínios (serviço, processo de software baseado na ISO e colaboração) a fim de criar catálogos de DROPs com aplicação da abordagem de catálogo de DROPs na construção de ontologias operacionais com apoio computacional da ferramenta. Com isso, foram implementados os catálogos de DROPs de cada um dos domínios. Como os domínios de serviço (S-OPL) (FALBO et

al., 2015a) e de processo de software baseado na ISO (ISP-OPL) (FALBO et al., 2015b) possuíam OPLs, também, nesta etapa, foi criada uma versão preliminar de uma OPL para o domínio de colaboração (C-OPL) (Capítulo 4, seção 4.5).

Avaliação da Aplicação da Abordagem de Catálogo com Suporte Automatizado - Os catálogos de DROPs foram aplicados com o apoio da ferramenta para a construção de algumas ontologias apresentadas em artigos e dissertações em que não se dispunha de um catálogo de DROPs e/ou suporte computacional para isso. Em Quirino et al. (2015), encontra-se a construção de uma subontologia de oferta de serviço; em Quirino et al. (2015), tem-se a construção de uma ontologia de requerimento do software e em Souza et al. (2016), uma ontologia de um editor de texto.

Escrita da Dissertação - Os resultados obtidos nas etapas anteriores foram descritos e utilizados nessa dissertação.

1.4 Validação

Para a validação das propostas feitas, catálogos de DROPs em três domínios distintos são criados no OLED com a utilização do novo módulo mostrando a flexibilidade do editor em trabalhar sem restrição quanto ao domínio. Esses domínios serão os de Serviço (S-OPL) (FALBO et al., 2016), de Processo de Software baseado na ISO (ISP-OPL) (RUY et al., 2015a) e de Colaboração (CONTO) (OLIVEIRA, 2009a). Os dois primeiros já se constituem em uma OPL, ou seja, apresentam um processo definido para a construção da ontologia. Já o último não o possui, assim, é uma contribuição desta dissertação uma proposta inicial desse processo, constituindo assim a primeira versão da C-OPL. O uso de diferentes catálogos nos possibilita a comparação entre os vários graus de maturidade deles, já que alguns destes são consolidados e maduros, enquanto o último está ainda em fase de construção.

Com esses catálogos criados, eles são utilizados para verificação da eficácia do novo módulo adicionado ao OLED em prover as funcionalidades de criação de catálogo de DROPs. Desta forma, pretende-se demonstrar que as novas funcionalidades apoiam de forma adequada a criação de catálogos de DROPs em qualquer domínio, desde que sejam utilizadas as classes de OntoUML implementadas no OLED.

Em seguida, os catálogos implementados são utilizados para a criação de algumas ontologias operacionais nos três domínios já mencionados para verificarmos a produtividade, praticidade e riqueza de informação presentes nas ontologias criadas. Nessas ontologias, busca-se exemplificar como a abordagem de desenvolvimento com reuso de DROPs pode solucionar ou minimizar diferentes problemas do desenvolvimento de ontologias.

Também foi realizada uma pesquisa para avaliar a abordagem metodológica pro-

posta para a criação de catálogos de DROPs e a utilização da abordagem de catálogos de DROPs para a modelagem de ontologias com suporte automatizado computacional para a criação e reutilização de catálogos de DROPs nessas modelagens.

1.5 Estruturação

Para fins de organização do trabalho realizado, nesta seção apresentamos a forma como está estruturado esta dissertação. No Capítulo 1 fazemos uma breve introdução do trabalho. Já no Capítulo 2, apresentaremos o embasamento teórico utilizado. No capítulo 3, apresentaremos alguns catálogos de DROPs em domínios distintos. O Capítulo 4 mostramos uma abordagem metodológica para extração de DROPs de uma ontologia de referência e a C-OPL. Enquanto no Capítulo 5, mostraremos a criação e utilização de catálogos de DROPs para a construção de ontologias operacionais por meio do reúso dos DROPs com suporte automatizado. Para o Capítulo 6 apresentamos alguns estudos de caso do uso do catálogo de DROPs. Então no Capítulo 7 comparamos algumas características de ferramentas e abordagens nelas implementadas em relação a construção de ontologias. Por último, temos o Capítulo 8 apresentando as considerações finais do trabalho realizado e algumas contribuições dele.

2 Base Teórica

2.1 Ontologia

O conceito de ontologia provavelmente se originou na Grécia antiga, como um objeto de estudo e reflexão de Platão e Aristóteles. Mesmo que a etimologia grega seja o registro mais antigo da palavra ontologia, é a sua forma latina que apareceu em 1606, na obra *Ogdoas Escolástica*, Jacob Loard (Lorhardus), e em 1613 o *Lexicon Philosophicum*, Rudolf Göckel sendo popularizado por Christian Wolff em seu trabalho *Philosophia prima sive Ontologia* em 1730 (GUIZZARDI, 2005; POIDEVIN et al., 2009). Para ontologia, qualquer coisa que exista é vista apenas como algo que é, em outras palavras, as preocupações ontológicas que determinam as categorias do ser são fundamentais, e perguntam em que sentido, os itens para estas categorias são ditos que "são".

O termo ontologia foi popularizado pelo filósofo alemão Christian Wolff, que definiu uma ontologia como a filosofia *raw* (filosofia primeira) ou ciência do ser como tal (HACKING, 2004). Assim, esta ciência teve um caráter racional e dedutivo, que teve como objetivo estudar as características mais gerais do ser. Várias áreas do conhecimento se utilizaram das ontologias em seus esforços para entendimento e estruturação do conhecimento, adaptando o conceito vindo da filosofia para os respectivos contextos.

Na literatura, são encontrados muitos estudos e definições para ontologia. Em Guarino (1995), ontologia pode ser vista como o estudo da organização e da natureza do mundo, independentemente da forma de nosso conhecimento sobre ele. Aqui o conceito tem sua definição ampliada com a noção de ontologia formal, ou seja, a teoria de distinções a priori entre as entidades do mundo (objetos físicos, os eventos, as regiões, as quantidades de matéria, etc), bem como entre as categorias metanível usadas para modelar o mundo (conceitos, propriedades, qualidades, estados, papéis, etc). Nesse contexto, a teoria das relações parte-todo e a topologia (teoria da relação de conexão) representam papéis fundamentais na criação de uma ontologia.

Segundo Gruber (1993), ontologia é uma especificação de uma conceituação. Conceituação, significando uma abstração, uma visão simplificada do mundo. Se a base de conhecimento de um sistema inteligente é representar o mundo para algum propósito, então, ele deve estar comprometido com alguma conceituação, explícita ou implicitamente. Mundo aqui se refere a algum fenômeno, tópico ou área sujeita. Já especificação significa uma representação formal e declarativa. Quando vista como um artefato usado para implementação, na estrutura de dados que representa a ontologia, os tipos dos conceitos utilizados e as restrições sobre uso deles são demonstrados de forma declarativa, explicita-

mente e usando uma linguagem formal. A representação formal implica que uma ontologia deve ser legível por máquina. No entanto, uma ontologia não pode ser executada como um programa. Ela representa algum conhecimento declarativo para ser usado por programas (GASEVIC; DJURIC; DEVEDZIC, 2009).

Para Kalfoglou (2001), ontologia é uma representação explícita de um entendimento comum de conceitos importantes em algum domínio de interesse. Todo o conhecimento individual é subjetivo; uma ontologia implementa uma estrutura cognitiva explícita que ajuda a mostrar objetividade como um acordo sobre a subjetividade. Não se supõe representar o conhecimento subjetivo de algum indivíduo, mas o conhecimento aceito por um grupo ou uma comunidade. Portanto o conceito de compartilhado significa aqui que uma ontologia captura algum conhecimento consensual. Logo uma ontologia transmite uma compreensão compartilhada de um domínio que é acordado entre um número de indivíduos ou agentes. Esse tipo de acordo facilita a comunicação precisa e eficaz de significado. Este, por sua vez, abre a possibilidade para o compartilhamento de conhecimento e reutilização, o que permite a interoperabilidade semântica entre agentes e aplicações inteligentes.

Já em Hendler (2001) ontologia é definida como um conjunto de termos de conhecimento, incluindo o vocabulário, as interconexões semânticas, bem como algumas regras simples de inferência e lógica para algum assunto particular. As partes importantes de definição são as interconexões semânticas, inferência e lógica. O artigo diz que uma ontologia especifica o significado das relações entre os conceitos utilizados. Além disso, ontologias permitem alguma forma de raciocínio. Por exemplo, uma ontologia sobre "Música" pode incluir conceitos sobre instrumentos e como tocá-los, bem como álbuns e como gravá-los (GASEVIC; DJURIC; DEVEDZIC, 2009), além de permitir inferências que descubram novo conhecimento não modelado diretamente.

Swartout e Tate (1999) apresentam uma definição muito interessante de ontologia que é a de uma estrutura básica ou armadura em torno do qual uma base de conhecimento pode ser construída. Como uma armadura, uma ontologia deve fornecer um "esqueleto de conhecimento" firme e estável ao qual todos os outros conhecimentos devem se referir, conforme ilustração na Figura 1. Em outras palavras, uma ontologia representa o conhecimento fundamental sobre um tema.

Da mesma forma que existem várias definições para ontologias, também existem vários tipos de ontologias (GUARINO, 1997). Uma das formas de diferenciar ontologias é pelo formalismo da expressividade das relações e do conteúdo conceitual de seus termos, ou seja, semântica que a classificação taxionômica possui. Como exemplos, citam-se as ontologias leves e pesadas (no inglês, *light weight* ou *heavy weight*, respectivamente). Outra classificação se relaciona ao nível de generalidade da natureza de aplicação. Como exemplos, têm-se as ontologias de domínio e de tarefa (ISOTANI; BITTENCOURT, 2015; CAMPOS et al., 2012). Em uma terceira perspectiva, pode-se classificar ontologias conforme a sua

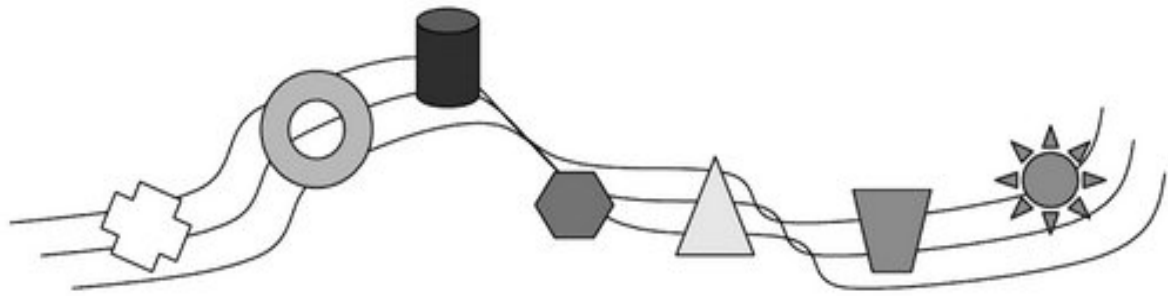


Figura 1 – Ilustração de uma ontologia segundo Swartout e Tate (GASEVIC; DJURIC; DEVEDZIC, 2009)

especificidade, sendo as de fundamentação as mais genéricas e as de domínio as mais específicas. Uma representação de classificação de ontologias de acordo com as dimensões expressividade, propósito e especificidade pode ser visto na Figura 2.

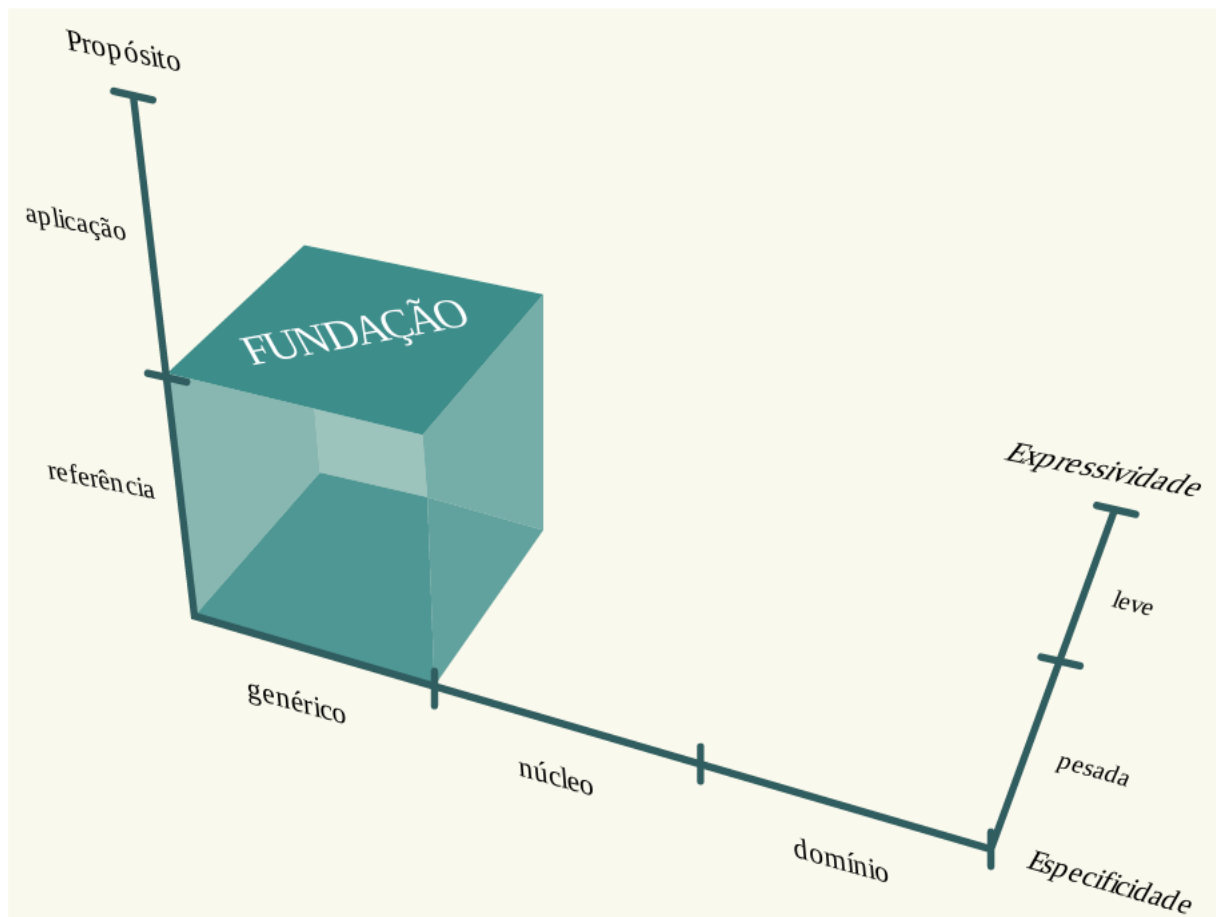


Figura 2 – Classificação de ontologias de acordo com a expressividade, propósito e especificidade (ISOTANI; BITTENCOURT, 2015)

Conforme (CAMPOS et al., 2012), ontologias leves são vocabulários simples, constituídos basicamente por uma hierarquia de termos e relações básicas, isto é, são aquelas que não se preocupam em definir detalhadamente cada conceito representado. A principal ênfase das ontologias leves é definir a taxonomia que representa a relação hierárquica entre

conceitos. Ontologias pesadas se preocupam não apenas a taxonomia, mas também com a representação rigorosa da semântica entre os conceitos por meio de axiomatização e restrição (SILVA et al., 2014). Em Isotani e Bittencourt (2015), tem-se que o desenvolvimento de ontologias pesadas requer uma formalização na definição de cada conceito, a organização desses conceitos baseados em princípios bem definidos, uma definição formal da semântica entre os conceitos e suas relações, além de outras considerações. Para criar bases de conhecimento reusáveis e compartilháveis, é fundamental definir ontologias pesadas.

Segundo Isotani e Bittencourt (2015), uma ontologia de tarefa é formada pela estrutura hierárquica que representa as restrições para executar ações básicas que são o resultado da decomposição dos passos de resolução de problemas, isto é, expressam conceitos sobre a resolução de problemas, ou seja, descrevem o vocabulário relacionado a uma atividade ou tarefa genérica (CAMPOS et al., 2012). As ontologias de domínio representam o conhecimento sobre um domínio específico, ou seja, define e caracteriza esse conhecimento, descrevem o vocabulário relacionado a um domínio genérico (SILVA et al., 2014). Conforme (ISOTANI; BITTENCOURT, 2015), a ontologia de domínio representa o conhecimento sobre um tópico, enquanto a ontologia de tarefa representa a habilidade de aplicar esse conhecimento para resolver problemas em diferentes situações. Essa distinção é muito importante, pois, por meio dela, torna-se possível criar sistemas e bases de conhecimento mais modulares, compartilháveis e extensíveis.

Dentro das ontologias de domínio, destacam-se as ontologias de referência e operacionais. Conforme Falbo (2011), uma ontologia de referência deve ser um tipo especial de modelo conceitual, um artefato de engenharia com a exigência adicional de representar um modelo de consenso dentro de uma comunidade. É uma especificação independente da solução, com o objetivo de fazer uma descrição clara e precisa de entidades de domínio para fins de comunicação, aprendizagem e resolução de problemas. Já uma ontologia operacional é uma versão de implementação de uma ontologia referência, ou seja, é a aplicação de uma ontologia de referência para a resolução de problemas naquele domínio, por exemplo da extensão das classes da ontologia de referência, por meio de uma fase de projeto que deve ser realizada levando em conta requisitos não funcionais, bem como a plataforma de implementação de ontologia.

Conforme as definições trazidas nesta seção, assim, uma ontologia deve ser formalmente representada, utilizando-se de uma linguagem (visual ou lógica), por meio de definições que devem ser explicitamente declaradas. Além disso, a conceituação que representam deve ser compartilhada por algum grupo de pessoas ou agentes de um domínio. Em suma, a ontologia fornece um vocabulário para descrever e raciocinar sobre os casos concretos de um domínio. Isso possibilita a interação entre indivíduos e comunidades através de ontologias, uma vez que compartilhem as conceituações por elas definidas, bem como usufruam de mecanismos de raciocínio automatizados via computadores.

2.2 OntoUML

Proposta por [Guizzardi \(2005\)](#), OntoUML consiste de um perfil ontologicamente bem fundamentado do diagrama de classes UML 2.0. A falta de uma definição semântica formal precisa da linguagem UML, podendo gerar inconsistências nos modelos gerados, foi minimizado com a extensão do metamodelo UML para que se tornasse isomórfico à UFO, suprimindo, assim, a carência de fundamentação formal e tornando-o ontologicamente consistente. A escolha de UML se deve às características presentes na linguagem como, por exemplo, os mecanismos de extensão que permitem modificar os elementos da linguagem para que um conjunto coerente de tais extensões constitua um perfil UML. Além disso, UML é vista como um padrão na área de Engenharia de Software, o que facilita sua adoção em contextos similares.

No perfil UML proposto, tem-se um conjunto de estereótipos, conforme mostra a Tabela 3, representando as categorias ontológicas dos tipos de universais propostos em UFO-A, bem como por restrições formais que refletem a axiomatização de UFO, de tal modo que se restringe o conjunto de modelos gramaticalmente válidos em OntoUML àqueles que representam situações admissíveis segundo a teoria da UFO ([GUIZZARDI, 2007](#)). Essa é uma teoria ontologicamente bem fundamentada e que foi utilizada para a concepção do meta-modelo de OntoUML.

Stereotype	Identity	Rigidity	Allowed Supertypes
Kind	Provide	Rigid	Category, RoleMixin, Mixin
Subkind	Carry	Rigid	Kind, Subkind, Category, RoleMixin, Mixin
Role	Carry	Anti-Rigid	Kind, Subkind, Role, Phase, Category, RoleMixin, Mixin
Phase	Carry	Anti-Rigid	Kind, Subkind, Role, Phase, Category, RoleMixin, Mixin
Category	-	Rigid	Category
Mixin	-	Semi-Rigid	Mixin, RoleMixin
Role Mixin	-	Anti-Rigid	Mixin, RoleMixin

Figura 3 – Distinção de tipos de objeto OntoUML ([RUY et al., 2015](#))

Por ser um perfil UML, de acordo com [Ruy et al. \(2015c\)](#), OntoUML permite fazer distinções de modelagem mais finas entre diferentes tipos de classes e relações, de acordo com as distinções ontológicas apresentadas pela UFO. OntoUML é, em si, uma linguagem baseada em padrões (embora independente de domínio), cujas primitivas de modelagem são padrões que incorporam as microteorias que compõem a ontologia de fundamentação UFO.

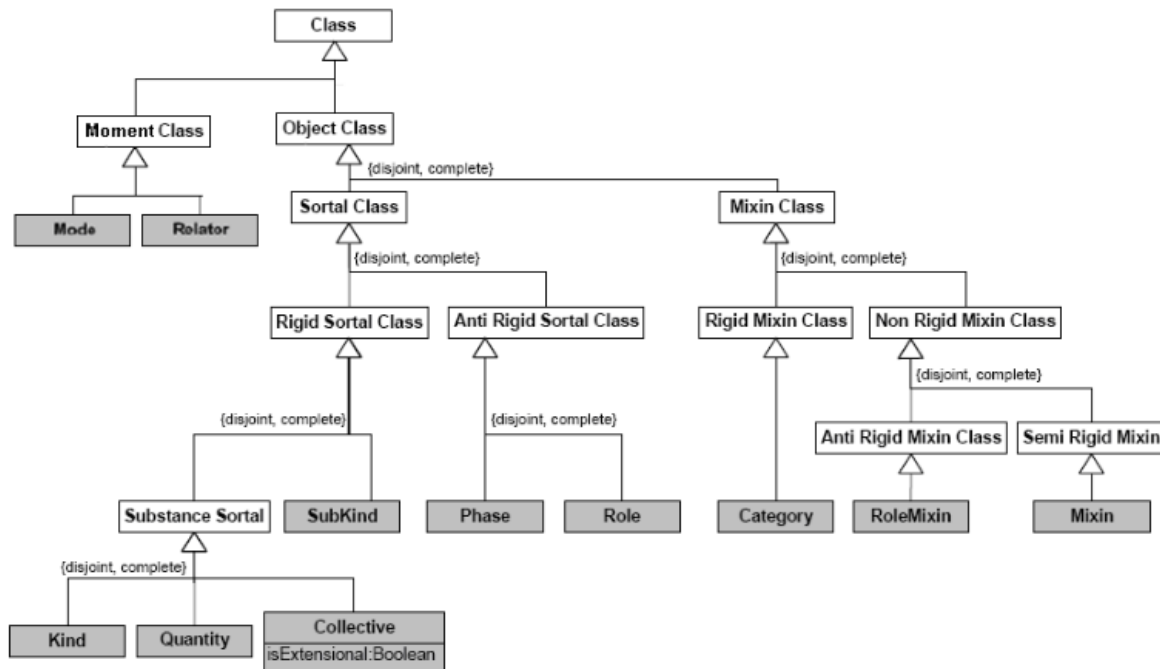


Figura 4 – Fragmento do metamodelo da linguagem OntoUML para o tipo *Class* adaptado (GUIZZARDI, 2005)

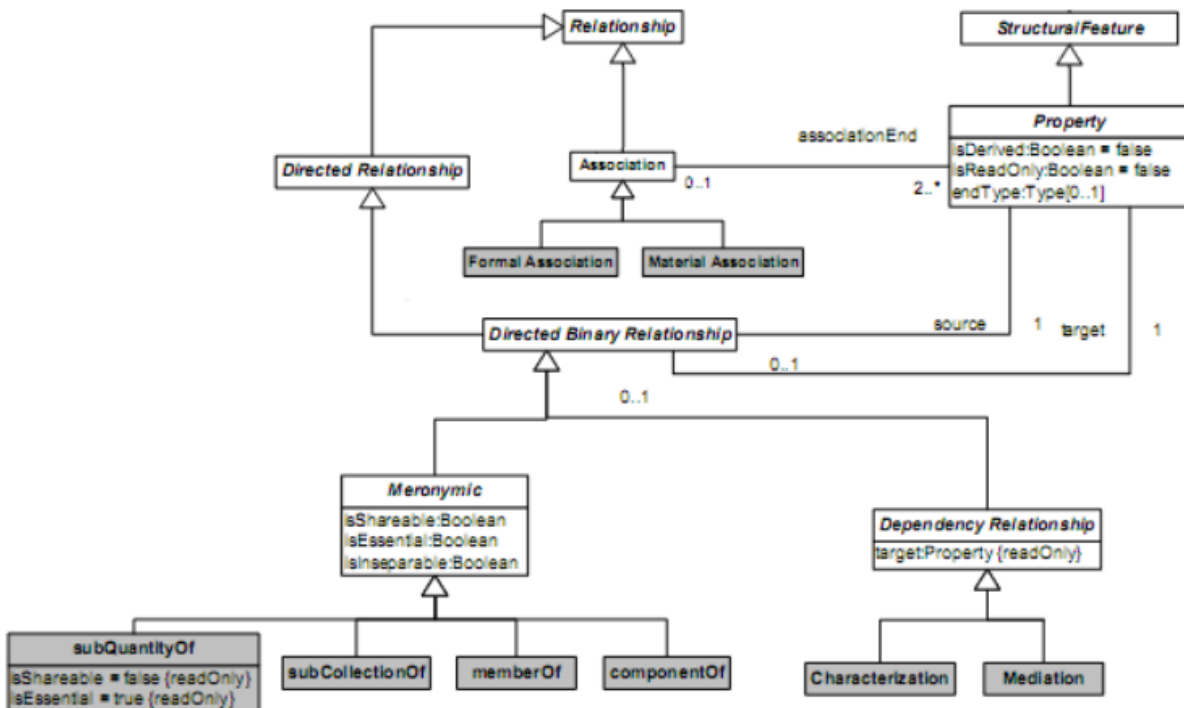


Figura 5 – Fragmento do metamodelo da linguagem OntoUML para o tipo *Relationship* adaptado (GUIZZARDI, 2005)

Devido ao grau de formalidade encontrado em *OntoUML* que propicia a construção de modelos precisos e consistentes, essa linguagem tem despertado muito interesse na comunidade acadêmica, na indústria e por instituições governamentais. Mesmo assim, com todo esse aparato de formalismo e fundamentação, a construção de modelos conceituais ontologicamente bem-fundamentados e, em especial, de ontologias de domínio bem-fundamentadas, é uma atividade de especial complexidade principalmente para modeladores iniciantes (GRAÇAS; GUIZZARDI, 2010).

Nas Figuras 4 e 5 pode-se ver fragmentos do metamodelo de *OntoUML* em que os elementos em cinza, ou seja, as folhas da árvore de elementos, representam as categorias de *OntoUML* (em outras palavras, os estereótipos). Em 4, o fragmento do metamodelo da *OntoUML* apresenta o elemento de representação *Class*, da UML, sendo especializado conforme os tipos de Universais Monádicos da UFO. Já em 5, o fragmento do metamodelo da *OntoUML* apresenta o elemento de representação *Relationship* sendo especializado conforme os tipos de Universais de Relação da UFO (GUIZZARDI, 2005; ZAMBORLINI, 2011). Essas categorias são as que são implementadas no OLED (ver seção 2.3) que oferece suporte automatizado a *OntoUML* e que foi usado como base do trabalho descrito nesta dissertação.

O foco de uma linguagem como *OntoUML* é a expressividade e clareza de representação dos elementos que a compõem, visando captar com precisão o conhecimento do domínio e todas as particularidades dele. Com isso, expressividade e clareza de representação são valorizadas em detrimento de outras características, tais como eficiência em processamento e decidibilidade. A ideia é que uma vez criada uma ontologia, esta, como modelo conceitual, possa então dar origem a diversas codificações, em diferentes linguagens (OWL DL, RDF, F-Logic, CASL) (GRAÇAS; GUIZZARDI, 2010). Conforme CMMi (2016), ampliando o entendimento, *OntoUML* apresenta as seguintes características:

- É uma notação para modelagem conceitual, um perfil UML para a modelagem ontológica bem fundamentada;
- É baseada na Ciência Cognitiva (conhecimento específico sobre nossa percepção) e sobre a lógica modal e os fundamentos matemáticos da lógica, conjuntos e relações. Construído sobre UFO;
- É criada a partir dos próprios fundamentos e constituem um sistema completo independente dos elementos UML originais. Ela usa alguns aspectos (como classes), no entanto, omite um conjunto de outros conceitos problemáticos (por exemplo, agregação e composição) e os substitui com os próprios conceitos ontologicamente corretos.

Apesar de não focalizar a implementação de ontologias, o uso correto de *OntoUML* pode favorecer uma implementação mais consistente e eficaz. Isso se dá porque, ao representar formalmente o domínio do sistema, pode-se gerar o código e esquemas de banco de dados produzindo um software que seja interoperável pelo modelo.

CMMi (2016), destaca alguns pontos fortes de OntoUML que são: (i) Capacidade de distinguir vários tipos de tipos de objetos; (ii) Capacidade de distinguir tipos de objetos rígidos e não rígidos; (iii) Capacidade de distinguir tipos de objetos dependentes e independentes; (iv) Capacidade de distinguir vários tipos de relacionamento todo-parte; (v) Capacidade de distinguir atributos de uma hierarquia de generalização e especialização; (vi) Baseado na lógica matemática estritamente definida e conhecimento da ciência cognitiva; (vii) Muito mais legíveis do que UML simples; (viii) Provada na modelagem conceitual de domínios de negócios.

Também segundo CMMi (2016), ainda existem alguns pontos em OntoUML que necessitam de mais evolução e contribuição para a disseminação dessa linguagem na modelagem de soluções. Alguns dos pontos fracos que OntoUML apresenta, são: (i) não é muito bem documentada ainda; (ii) não há muitos praticantes desta linguagem; (iii) há falta de ferramentas OntoUML; (iv) não foi ainda provada em um processo de desenvolvimento de um software real.

Mesmo assim, conforme Quirino et al. (2015), usando OntoUML, é possível contar com um aparelho de engenharia de ontologia bem mantido que pode ser aplicado para construir ontologias de domínio relacionadas a serviços. Esse aparelho inclui a verificação e validação de modelos através de simulação visual, bem como a transformação de modelos em linguagens como OWL e suporte para aplicações de padrões de ontologia. Essa é uma característica muito importante para esta pesquisa, para uso na proposta da abordagem de catálogo de DROPs para construção de ontologias operacionais de domínio. OntoUML é usada para representar os padrões porque é um perfil UML que permite fazer distinções de modelagem mais finas entre diferentes tipos de classes e relações de acordo com as distinções ontológicas apresentadas pela ontologia de fundamentação UFO-A, uma ontologia de *endurants*, ou seja, de tipos duradouros independentes de domínios segundo Falbo et al. (2014).

2.3 OLED - OntoUML Lightweight Editor

O OLED (*OntoUML Lightweight Editor*) é uma ferramenta desenvolvida para apoiar a Engenharia de Ontologias nas tarefas de formalização, verificação, validação e implementação, que são muito importantes na tarefa de construção de ontologias. Ela implementa o metamodelo de referência OntoUML (CARRARETTO, 2010; ALBUQUERQUE; GUIZZARDI, 2013), permitindo o uso das categorias de OntoUML por meio das classes disponibilizadas para a modelagem de ontologias. Em outras palavras, o OLED oferece um ambiente para o desenvolvimento, avaliação e implementação de ontologias de domínio, usando a linguagem de modelagem ontologicamente fundamentada com base em UFO OntoUML.

Para atender aos objetivos citados no parágrafo acima, o OLED agrega tecnologias desenvolvidas para OntoUML em projetos de pesquisa realizados pelo Núcleo de Estudos em Modelagem Conceitual e Ontologias (NEMO). Nele, estão agregados resultados de pesquisas relacionadas com a OntoUML disponibilizando uma série de funcionalidades adicionais, que apoiam os usuários ao longo do desenvolvimento, por exemplo, (i) erificação sintática automática, obtida a partir da implementação da RefOntoUML; (ii) suporte para a especificação, verificação e validação de restrições OCL (GUERSON; ALMEIDA; GUIZZARDI, 2014); (iii) transformações alternativas de OntoUML para OWL (BARCELOS et al., 2013); (iv) validação do modelo através de simulação visual e gerenciamento de anti-padrão (SALES, 2014; GUERSON et al., 2015).

O OLED é uma ferramenta multiplataforma, ou seja, é possível executá-lo nos sistemas operacionais mais comuns (Mac OS, Windows e Linux) e é distribuído como um executável java (.jar). A ferramenta é baseada principalmente no editor TinyUML e agrega muitas contribuições diretas e indiretas para os membros do grupo de pesquisa NEMO. Nele, também existe suporte para importar modelos de ontologias feitos no Enterprise Architect (EA) da Sparx Systems ¹, em que se pode importar modelos concebidos com a EA no OLED e se beneficiar de todas as capacidades do editor NEMO (2016).

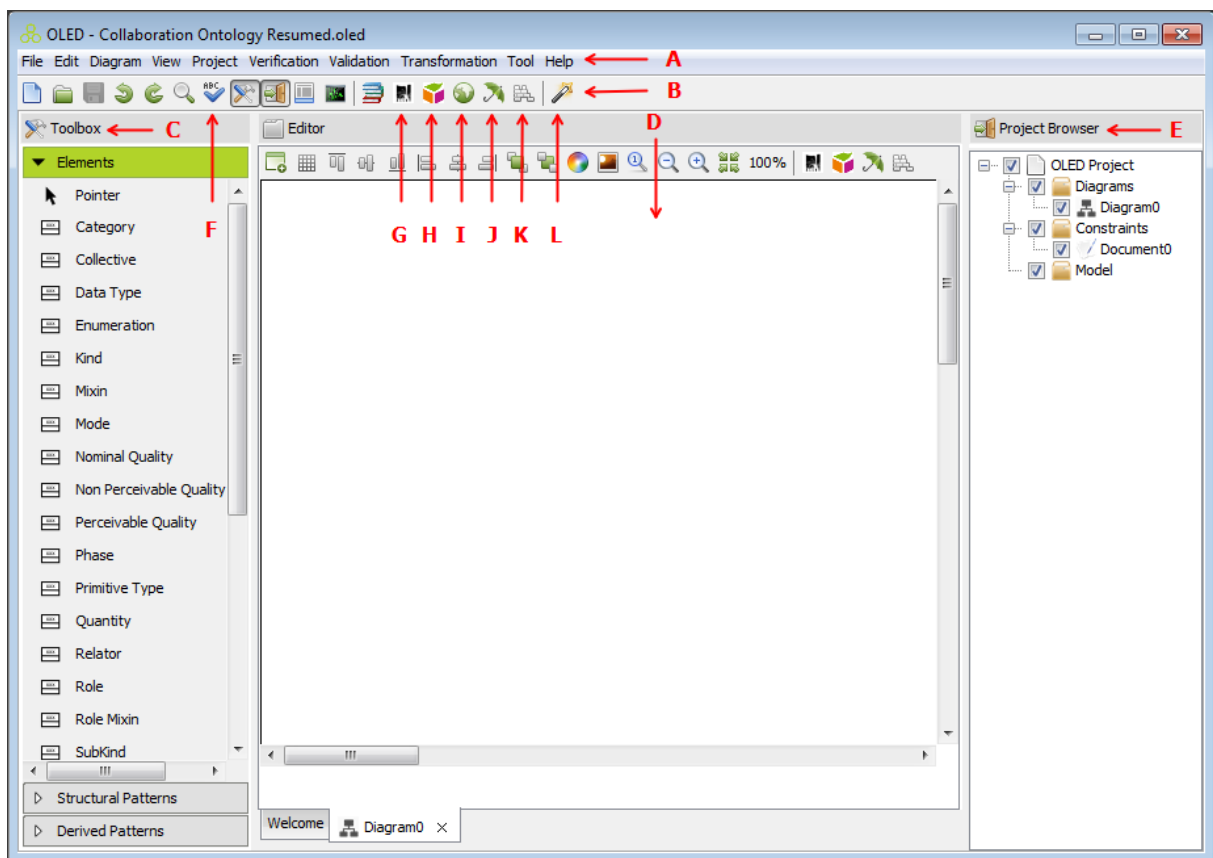


Figura 6 – Tela de um projeto novo criado no OLED

¹ <http://www.sparxsystems.com.au/>

A Figura 6 apresenta a tela inicial do OLED, com um projeto novo criado. Nela, na parte superior da interface de OLED, indicado pelo rótulo *A*, existem as opções do menu principal, como Arquivo, Editar, Diagrama, Visão, Projeto etc., juntamente com os respectivos submenus. Debaixo do menu principal, existe a barra de ferramentas com atalhos para as funções mais usadas, indicados pelo rótulo *B*. Verticalmente, no lado esquerdo, podem ser vistas: (i) a caixa de ferramentas, com as categorias OntoUML; (ii) a biblioteca de padrões ontológicos estruturais; (iii) a biblioteca de padrões ontológicos de derivados (indicado pelo rótulo *C*. Rotulada com *D*, no centro, existe uma área vazia, onde um modelo OntoUML pode ser criada e editado. Rotulada com *E*, há o navegador de projeto, que mostra a estrutura do projeto, disposto em uma árvore, com suas subdivisões: Diagramas, Restrições e Modelo.

Na barra de ferramentas rotulado com *B*, existem alguns botões de atalho, referentes às outras funcionalidades disponíveis no OLED (rotulados conforme a seguir): *F* - a verificação semântica; simulação Gmodel utilizando um analisador *Alloy*, que permite que o engenheiro de ontologias examinar instâncias do modelo editado para verificar a consistência de sua ontologia; *H* - geração de código de ontologia usando OWL/SWRL; *I* - transformação da ontologia em um documento na linguagem de verbalização SBVR; *J* - verificação de anti-padrões ontológicos, que permite que o engenheiro de ontologias corrija erros de modelagem da ontologia; *K* - avaliação ontológica, sobre a transitividade das relações meronímicas e todo-parte; e, finalmente, *L* - avaliação ontológica sobre a integralidade.

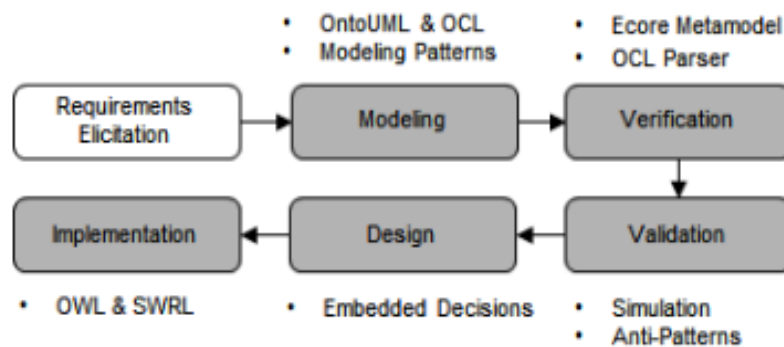


Figura 7 – Suporte do OLED ao processo de desenvolvimento de ontologias (GUERSON et al., 2015)

Os engenheiros de ontologias podem especificar suas ontologias de domínio com OntoUML, restringindo-as usando a *Object Constraint Language* (OCL²) para expressar classes invariantes de UML e regras de derivação para pontos finais e atributos de associação UML. A ferramenta fornece um conjunto de padrões de projeto integrados para acelerar a atividade de modelagem através da reutilização. Para melhorar a qualidade dos modelos construídos usando OLED, ele fornece uma verificação de sintaxe automática ao lado de

² <http://www.omg.org/spec/OCL/>

dois recursos de validação complementares, simulação visual e anti-padrões. Para aplicar o conhecimento formalizado no OntoUML em aplicações de web semântica, OLED apresenta uma série de transformações automáticas predefinidas para a *Web Ontology Language* (OWL) (possivelmente reforçada com regras SWRL) (GUERSON et al., 2015).

A Figura 7 ilustra as fases do desenvolvimento de ontologias usando OntoUML. A fase *Requirements Elicitations*, em que são levantados os requisitos, ainda não é suportada no OLED mas se faz necessária para o processo. A fase *Modeling* tem como principais funcionalidades a modelagem em OntoUML, com auxílio de OCL e modelagem de padrões. Para a fase *Verification* as principais funcionalidades se relacionam ao metamodelo ECore, que incorpora um conjunto de regras sintáticas para refletir axiomatizações formais do UFO, bem como o verificador OCL. O OLED tem a capacidade de verificar automaticamente se um determinado modelo está em linha com estas restrições formais, apontando exatamente quais construções no modelo estão em desacordo com elas (GUERSON et al., 2015).

Ainda na Figura 7, para a fase de *Validation*, há as funcionalidades de simulação e verificação de anti-padrões, que consistem em analisar se o modelo formaliza precisamente a conceituação compartilhada do domínio. O OLED fornece simulação visual e gestão de anti-padrões ontológicos. A fase de *Design* tem como principal funcionalidade as inferências/deduções para descobrir conhecimento e anotá-lo semanticamente. E para a fase *Implementation*, estão disponíveis as funcionalidades de transformação do modelo em OntoUML para OWL e SWRL. OWL é uma extensão do RDF baseado em lógica descritiva para representar o conteúdo no contexto da Web Semântica. Ela tem sido usada para implementar as ontologias de referência para descobrir o conhecimento, anotar semanticamente seu conteúdo e para publicá-lo na web. SWRL é a linguagem usada para expressar regras lógicas sobre especificações OWL (GUERSON et al., 2015).

Como resultado do presente trabalho, o OLED será dotado de funcionalidades que possibilitam a aplicação da abordagem de catálogos de DROPs, com: a) a criação e a utilização de catálogos de DROPs; e b) a aplicação da reutilização desses padrões na construção de ontologias operacionais de domínio. Dessa forma, uma ontologia será desenvolvida com apoio de ferramenta computacional, ganhando ainda mais agilidade e produtividade e, conseqüentemente, levando a resultados com maior qualidade.

Em linhas gerais, os catálogos de DROPs são concebidos e implementados como projetos OLED, em que cada potencial padrão corresponde a um diagrama neste projeto. Potencial porque, neste caso, a interpretação do diagrama como um OP depende de configurações que são realizadas no projeto para que o OLED, assim, o identifique. Caso o OLED identifique um projeto como um catálogo de DROPs, o mesmo é carregado e torna-se disponível na caixa de ferramentas localizada no lado esquerdo da interface de OLED. Mais detalhes de como se criar um catálogo de padrões ontológicos, bem como a sua utilização, são apresentados no capítulo 5.

2.4 Padrões Ontológicos

Intuitivamente, tem-se que o termo padrão remete ao sentido de modelo, regra a ser seguida, estrutura básica, armadura ou algo que produz resultado previsto, ou seja, remete a previsibilidade. Os padrões estabelecem diretrizes mínimas que comportamentos, atividades e produtos devem possuir ou a que devem se ajustar para serem eficazes, úteis ou confiáveis. Portanto, um padrão é algo que pode ser reconhecido com regularidade no mundo ou em um projeto, porque os elementos em um padrão repetem-se de maneira previsível.

A história mostra vários especialistas que trabalharam em pesquisas para a identificação de padrões, por exemplo, nas guerras. Esses profissionais tinham a tarefa de reconhecer padrões dos inimigos em suas comunicações, ações e estratégias. O que eles buscavam com os trabalhos deles era a identificação de regularidades que, se entendidas, se transformam em informações estratégicas. Esses padrões, então, eram e são utilizados, por exemplo, para comparação com outros padrões, aprendizagem de máquina, análise de texto, criptografia e descritografia de informação etc. (LLOYD, 2013).

Com o avanço computacional e as pesquisas científicas, muitas áreas têm-se beneficiado do reconhecimento de padrões. Conforme Blomqvist (2009), um exemplo é a área de visão computacional, em que um programa é utilizado para reconhecimento de padrões de objetos e, mais recentemente, a automação de veículos que se utiliza do reconhecimento de padrões para a detecção de obstáculos, estradas, placas, sinais de trânsito, além de outras coisas. Outros esforços têm sido dedicados em dotar computadores e sistemas de alguma capacidade de aprendizagem, respondendo a estímulos externos, ou seja, identificando padrões na interação com o mundo exterior e respondendo a esses padrões.

Outro exemplo do uso de reconhecimento padrões é na recuperação de informação (IR) e zonas de extração de informação (IE). IR é uma área que se desenvolveu rapidamente, juntamente com o desenvolvimento web como uma grande fonte de informação disponível ao redor do mundo. Exemplos de aplicações IR pode ser para automatizar busca e personalização de conteúdo na web. Assim, um novo tipo de modalidade de busca, chamada busca semântica surgiu. A busca semântica é aquela que usa o reconhecimento de padrões e os padrões do gráfico, para além de palavras-chave, para obter resultados melhores (BLOMQVIST, 2009).

Com as ontologias não foi diferente. O interesse em encontrar padrões e utilizá-los para resolução de problemas em modelagens vêm ganhando força nos últimos anos. Segundo Blomqvist (2009), na perspectiva de reconhecimento de padrões, padrões são conjuntos recorrentes de entidades que podem ser encontrados em alguns conjuntos de estruturas, por exemplo, conjuntos recorrentes de conceitos e relações em ontologias. Neste sentido, não há nenhuma exigência sobre o que um padrão pode conter, o seu

nível de abstração, ou como ele pode ser estruturado ou recuperado. Por outro lado, padrões podem ser vistos como modelos de perspectiva, já que os padrões comumente são modelos cuidadosamente elaborados e que representam uma perspectiva de consenso sobre como resolver um problema específico em um domínio específico. Os modelos têm de ser suficientemente gerais e abstratos, a fim de serem reutilizáveis em um maior número de casos possível. Além disso, eles têm que representar a maneira comumente aceita para resolver o problema por uma comunidade.

A busca por padrões, estruturas ou conceitos que se repetem é de suma importância, conforme [Falbo et al. \(2013a\)](#), porque padrões são veículos para encapsular conhecimento e são considerados um dos meios mais eficazes para a nomeação, organização e raciocínio sobre o conhecimento. Também, de acordo com ([BUSCHMANN; HENNEY; SCHMIDT, 2007](#)), um padrão descreve um problema recorrente de modelagem especial, que surge em contextos específicos do projeto e apresenta uma solução bem provada para o problema. A solução é especificada, descrevendo o papel de seus participantes constituintes, suas respectivas responsabilidades e relacionamentos, e as maneiras pelas quais eles colaboram.

Sem a utilização de OPs, conforme [Quirino \(2016\)](#), um engenheiro de ontologias que deseja desenvolver uma ontologia reutilizando outra, tipicamente, realiza buscas em um conjunto de ontologias leves e as avalia em relação à tarefa ou ao domínio de interesse, ou utiliza ontologias de referência ou de núcleo que devem ser diretamente reutilizadas e especializadas. No primeiro caso, é necessário um processo de adaptação para lidar com requisitos da nova ontologia, o que pode ser custoso. No segundo caso, mesmo que as ontologias tenham sido bem projetadas, elas usualmente são grandes e abrangem mais conhecimento do que o necessário ao engenheiro de ontologias, sendo difícil identificar as partes realmente úteis e reutilizar apenas essas partes. Percebe-se, com isso, que a tarefa de reutilização conta com a subjetividade de quem seleciona e avalia as ontologias, o que pode comprometer a qualidade do resultado. O fato das ontologias selecionadas para reutilização serem muito genéricas ou muito grandes dificulta a seleção de quais partes serão reutilizadas, tornando o trabalho árduo e comprometendo o resultado.

Na Figura 8, podem ser vistos os vários tipos de OPs, segundo ([FALBO et al., 2013b](#)).

Dos vários tipos de OPs apresentados na Figura 8, o interesse desta dissertação perpassa pelos Padrões Ontológicos Conceituais (COPs) uma vez que são fragmentos de ontologias de fundamentação (FOPs) ou ontologias núcleo/domínio (DROPs), foco deste trabalho. Os COPs são utilizados durante a fase de modelagem conceitual da ontologia, sem preocupação com qualquer aspecto que não seja da modelagem, ou seja, tecnologia ou linguagem a ser usada para implementação da ontologia ([FALBO et al., 2016](#)).

A Figura 9 mostra os níveis de generalidade e os tipos de padrões ontológicos associados, FOP e DROP.

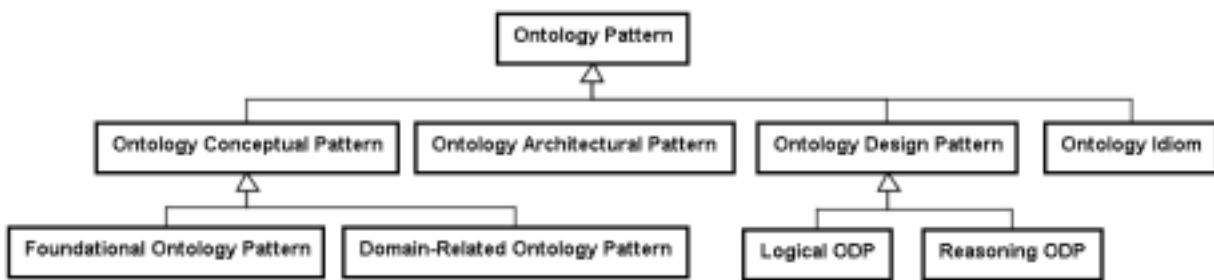


Figura 8 – Tipos de padrões ontológicos (FALBO et al., 2013b)

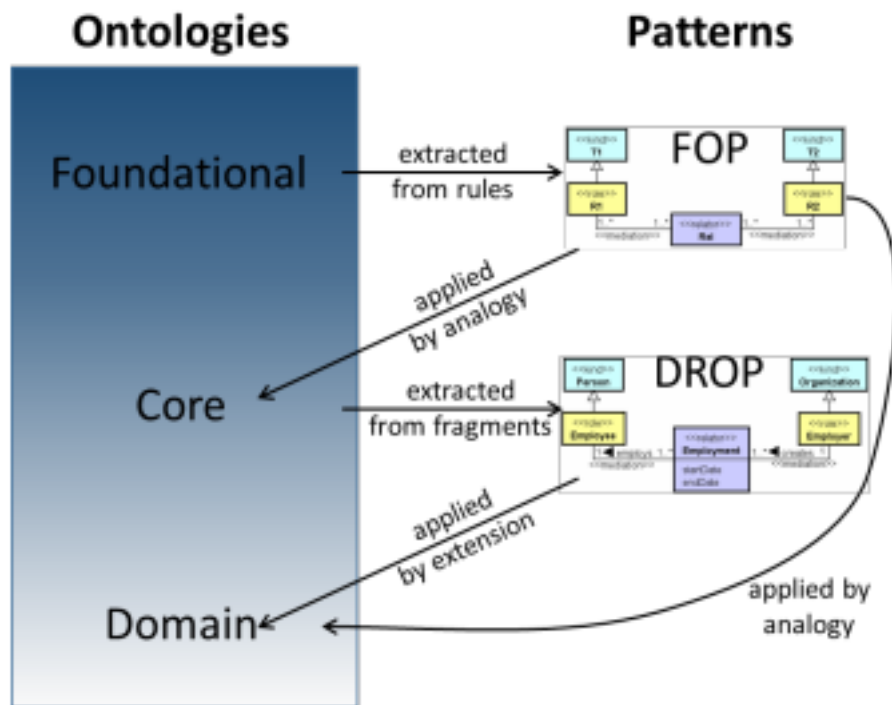


Figura 9 – Níveis de generalidade e padrões ontológicos (RUY et al., 2015)

FOPs são fragmentos reusáveis extraídos das regras das ontologias de fundamentação/núcleo e uma vez que as ontologias de fundamentação definem conceitos genéricos e independentes de domínio, FOPs possuem a capacidade de poderem ser usados em quaisquer domínios (FALBO et al., 2013b; GAILLY; ROELEN; GUIZZARDI, 2016). De acordo com Ruy et al. (2015), um FOP, capturando conteúdo estrutural de fundamentação, pode ser reutilizado por analogia, reproduzindo sua estrutura na ontologia em desenvolvimento, combinando os conceitos referenciados do padrão aos correspondentes no domínio. O resultado é um fragmento de ontologia com a estrutura FOP moldando as estruturas ao nível de conceitos de domínio. As FOPs podem ser aplicadas para construir ontologias de núcleo e de domínio, conforme lustrado na Figura 9.

Já os DROPs capturam o conhecimento básico do domínio de destino e podem ser extraídos diretamente de fragmentos de ontologias de núcleo/domínio, vide Figura 9. Em Ruy et al. (2015), tem-se que a reutilização de DROPs ocorre por extensão, isto é, seus

conceitos e relações se tornam parte da ontologia de domínio e são tipicamente estendidos, dando origem a novos conceitos e relações mais específicos. Ainda é reforçado que quando um DROP é extraído de uma ontologia de núcleo/domínio modelada já reutilizando FOPs resulta em uma DROP mais rica, portadora de conhecimento estrutural e de domínio, caracterizando uma aplicação de um padrão ontológico conceitual encadeada no nível de domínio.

Conforme disposto, tanto os DROPs como os FOPs podem ser aplicados pela Engenharia de Ontologias de domínio. Neste trabalho, o foco será na reutilização de DROPs com a criação e utilização de catálogos que serão apoiados por uma ferramenta, com a capacidade de gerenciamento de catálogo de DROPs. Exemplos de DROPs podem ser encontrados no Capítulo 3 desta dissertação.

2.5 OPLs - Linguagens de Padrões Ontológicos

Com o avanço das pesquisas relacionadas aos DROPs, foram desenvolvidas abordagens de utilização e reutilização dos mesmos para a construção de ontologias, das quais citamos as OPLs. Segundo Falbo et al. (2013a), OPL é uma rede de padrões ontológicos inter-relacionados ao domínio, que oferece suporte holístico para resolver problemas de desenvolvimento de ontologias para um domínio específico. Assim sendo, uma OPL agrupa DROPs de domínios específicos, que fornecem soluções recorrentes de modelagem para apoiar o engenheiro de ontologias na tarefa de modelagem. Outra característica das OPLs é disponibilizar um processo que fornece orientação explícita sobre quais problemas podem surgir nesse domínio, informando a ordem para resolver esses problemas e sugerindo um ou mais padrões para resolver cada problema específico (FALBO et al., 2015a).

Uma OPL se assemelha a um método de construção de ontologias, pois fornece um processo que descreve como usar os padrões para resolver problemas relacionados a um domínio específico. Apesar dessa semelhança, Falbo et al. (2013a) enfatiza que uma OPL não é um método para construir ontologias. Trata-se apenas da reutilização no desenvolvimento da ontologia e sua orientação pode ser seguida por engenheiros de ontologia usando qualquer método de desenvolvimento de ontologia que considere a reutilização de ontologia como uma das suas atividades.

Um exemplo de OPL, é o que é apresentado na Figura 10, que apresenta o modelo do processo da S-OPL, linguagem de padrões ontológicos no domínio de serviço, cuja notação utilizada é adaptada da notação para diagramas de atividade UML (FALBO et al., 2015a).

A construção de OPL implica na definição de DROPs e um processo que guia as relações entre eles. A Figura 10 traz, ainda, a simbologia utilizada na construção do processo da OPL, com os respectivos significados. Em OPLs, há basicamente três de tipos

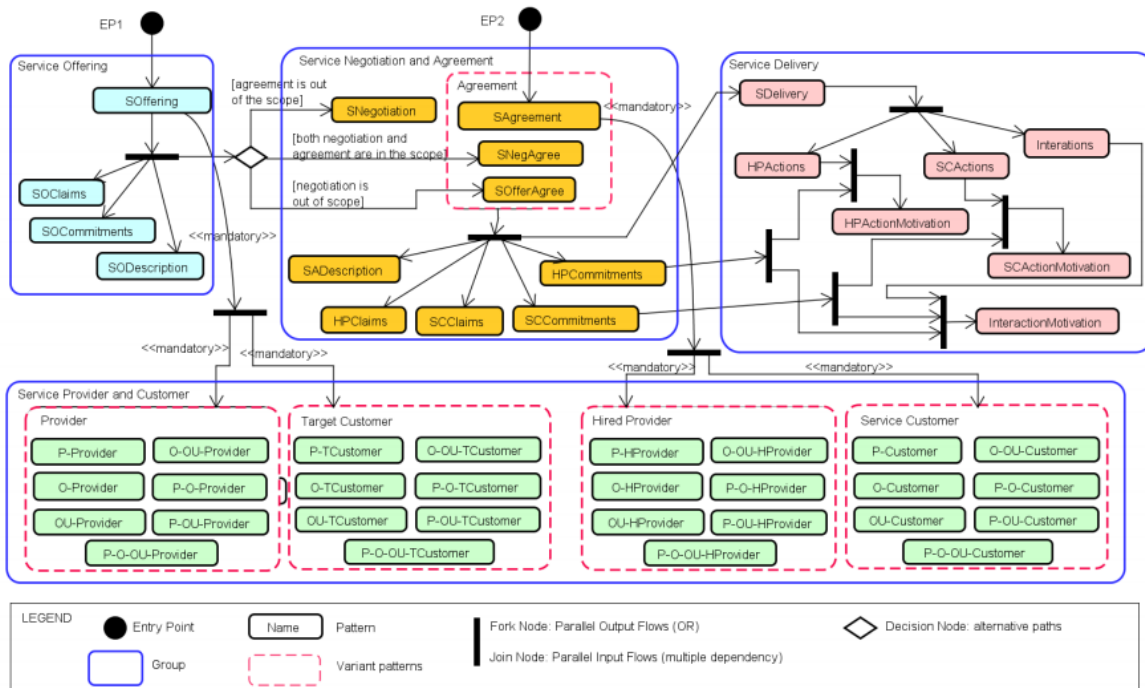


Figura 10 – Modelo do processo da S-OPL (FALBO et al., 2015a)

de relação entre os DROPs: (i) alternativa, em que um DROP pode substituir outro na modelagem; (ii) complemento, em que um DROPs complementa outro e (iii) agrupamento, em que DROPs são agrupados por afinidade na solução. As múltiplas relações que podem existir entre padrões ajudam a fortalecer e ampliar o poder de um padrão individual para além do foco específico (BUSCHMANN; HENNEY; SCHMIDT, 2007).

Em suma, uma OPL deve indicar claramente quais padrões referenciados abordam aspectos obrigatórios e quais abordam aspectos opcionais. Os padrões referenciados devem ser apresentados na ordem da aplicação sugerida. Sem essa orientação processual explícita, uma representação que se encaixa na rede básica dos padrões pode não fornecer um processo adequado que ajude a garantir uma ontologia suficientemente completa e bem formada (FALBO et al., 2015b). É interessante destacar o encadeamento do conhecimento entre as OPLs, OPs e ontologias. Uma sendo construídas com o suporte oferecido pela outra, começando da última para a primeira. E no aspeto da modelagem, OntoUML servindo como uma linguagem de modelagem.

3 Catálogo de Padrões Ontológicos de Domínio

Conforme [Falbo et al. \(2016\)](#), as abordagens de Engenharia de Ontologias orientadas por padrões têm sido reconhecidas como promissoras para lidar adequadamente com o desenvolvimento ontológico. Um OP descreve um problema de modelagem recorrente específico, que surge em contextos de desenvolvimento de ontologias e apresenta uma solução para esse problema. O uso de OPs contribui para a reutilização de modelos conceituais já consolidados, agilizando o processo de construção de novas ontologias, contribuindo, assim, para melhorar a comunicação, evitando ambiguidades com relação a definições e restrições dentro de um domínio, facilitando a captura, representação e reuso do conhecimento.

Nesse sentido, [Souza et al. \(2016\)](#) também argumenta que a reutilização tem sido amplamente defendida para ontologias a serem construídas com base em modelos preexistentes, acelerando o processo de desenvolvimento e levando a resultados de melhor qualidade. Por isso os OPs têm sido apontados como ferramentas interessantes para permitir essa reutilização. E para que sejam utilizados, esses OPs vêm sendo organizados em catálogos, que possibilitam a adição de estrutura às coleções de OPs ([SILVA, 2008](#)).

Em uma analogia, poderíamos comparar os catálogos de OPs Relacionados a Domínios (DROPs) com caixas de ferramentas. Uma caixa de ferramentas possui ferramentas selecionadas para um fim específico, ou seja, existem diversos tipos de caixas de ferramentas criadas para atender as necessidades de áreas específicas de atuação. Por exemplo, uma caixa de ferramentas de um bombeiro hidráulico será diferente da de um electricista porque, em cada uma delas, existem ferramentas para apoiar o profissional no desempenho do trabalho em sua própria área de atuação. Com os catálogos de DROPs não é diferente. Vários catálogos de DROPs podem ser criados para atender ao engenheiro de ontologia na modelagem de ontologias nos mais variados domínios. Em uma analogia direta, uma caixa de ferramenta está para um catálogo de DROPs assim como as ferramentas presentes na caixa de ferramentas estão para os DROPs contidos no catálogo.

Diante disso, as subseções seguintes apresentam algumas coleções de DROPs, desenvolvidos para domínios distintos ([FALBO et al., 2015a](#); [FALBO et al., 2015b](#); [OLIVEIRA, 2009a](#)). Essas coleções foram especificadas em documentos denominados Documentos de Especificação de Padrões Ontológicos, não possuindo, até o desenvolvimento desta dissertação, uma versão de uso automatizado.

3.1 Domínio de Serviço

Serviço é uma das áreas da economia que mais cresce, tornando-se uma das maiores. Muitos modelos empresariais de negócio são baseados em serviços. Conforme Falbo et al. (2016), essas empresas precisam representar e compartilhar conhecimento sobre seus modelos de serviço e as ontologias são instrumentos úteis para a representação e compartilhamento do conhecimento. Essa característica é de fundamental importância para empresas que querem prover serviços com cada vez mais qualidade, integrar esses serviços a outros e se destacarem no mercado. Esforços têm sido realizados na criação de meios que venham apoiar a construção de ontologias, neste caso, no domínio de serviço. Nesse sentido, foram realizados esforços para criação de uma OPL de serviço para apoiar a criação de ontologias do domínio de serviço (QUIRINO et al., 2015; FALBO et al., 2016).

Uma OPL é um conjunto de OPs de um domínio específico, inter-relacionados, que modelam problemas e soluções recorrentes deles e, dessa forma, favorecendo a reutilização. Conforme encontramos em Falbo et al. (2015a), uma OPL tem como objetivo fornecer suporte holístico para o uso de DROPs no desenvolvimento de uma ontologia de domínio. Ele fornece orientação explícita sobre quais problemas de modelagem podem surgir nesse domínio, informa a ordem para resolver esses problemas e sugere um ou mais padrões para resolver cada problema específico. Além disso, uma OPL considera explicitamente combinações de padrões para resolver um dado problema e conflitos entre eles, juntamente com diretrizes para integrá-los em um modelo conceitual de ontologia concreto.

Os padrões da S-OPL foram extraídos do UFO-S, uma ontologia de referência básica baseada em compromissos para serviços (NARDI et al., 2015), cuja conceituação se baseia no estabelecimento e cumprimento de compromissos e reivindicações entre participantes de serviço, prestadores ou clientes, ao longo do ciclo de vida do serviço. Falbo et al. (2015a) argumenta que UFO-S foca nas três principais fases do ciclo de vida do serviço que foram nomeadas como: oferta de serviços, negociação de serviços e prestação de serviços. Logo, os padrões da S-OPL são organizados em quatro grupos: Oferta de Serviços, Negociação de Serviços e Contrato, Prestação de Serviços, Provedor de Serviços e Cliente. Como um OPL, além dos grupos de padrões, o S-OPL fornece um processo para guiar o uso dos mesmos.

O interesse desta pesquisa é no conjunto de DROPs definidos para a S-OPL (QUIRINO et al., 2015; FALBO et al., 2015a; FALBO et al., 2016). O objetivo é estruturar e organizar os DROPs da S-OPL, na forma de catálogo de DROPs, em uma ferramenta computacional (OLED) para apoiar os engenheiros de ontologias na construção de ontologias operacionais de serviço, inclusive servindo de apoio para a própria S-OPL, uma vez que com o OLED será possível criar e utilizar catálogos de DROPs que são a base das OPLs.

¹ <<https://nemo.inf.ufes.br/wp-content/uploads/2015/12/s-opl-v1.5.pdf>>

3.1.1 Padrões Ontológicos de Serviço

Os padrões ontológicos de serviço¹ aqui apresentados são provenientes da S-OPL (QUIRINO et al., 2015; FALBO et al., 2015a; FALBO et al., 2016), uma linguagem de padrões ontológicos no domínio de serviço, que foram extraídos do UFO-S, uma ontologia de referência básica baseada em compromissos para serviços, cuja conceituação se baseia no estabelecimento e cumprimento de compromissos e reivindicações entre os participantes do serviço (prestadores de serviços e clientes de serviços) ao longo do ciclo de vida do serviço (NARDI et al., 2015). A seguir, é apresentada a definição parcial de alguns padrões da S-OPL que serão utilizados para validar a abordagem de catálogo de padrões proposta neste trabalho. A especificação completa pode ser vista no documento de especificação da S-OPL (FALBO et al., 2015a).

3.1.1.1 O-Provider - *Organization Provider*

Este padrão representa as organizações (*Organization*) como provedoras de serviço (*Service Provider*). Uma organização (*Organization*) é definida como um agente social que tem como objetivo principal a prestação de serviços ou produtos, e que é formalmente reconhecido pelo seu ambiente externo. Já o provedor de serviço (*Service Provider*) é o papel desempenhado pela organização (*Organization*) quando se compromete com uma comunidade cliente alvo (*Target Costumer Community*) por meio de uma oferta de serviço (*Service Offering*) (FALBO et al., 2015a). A Tabela 1 apresenta algumas características do padrão e a Figura 11 apresenta o modelo conceitual.

Tabela 1 – Especificação parcial do padrão *Organization Provider* (O-Provider) (FALBO et al., 2015a) adaptado*

Característica	Descrição
Name/Achronym	Organization Provider (O-Provider)
Alternative Patterns	
Intent	Represents organizations as <i>Service Providers</i> .
Rationale	<i>Organizations</i> can act as (play the role of) <i>Service Providers</i> , i.e., the ones responsible for the service provision.
Competency Questions	- Which are the types of service providers?

* Apesar desta dissertação ser escrita em português, as especificações serão mantidas em inglês por ser a língua original da especificação dos padrões

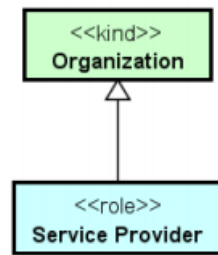


Figura 11 – Modelo conceitual do padrão *Organization Provider* (O-Provider) (FALBO et al., 2015a)

3.1.1.2 SOffering - *Service Offering*

Para que um serviço seja prestado, é necessário um provedor de serviços (*Service Provider*) que é o papel desempenhado pelos agentes quando esses agentes se comprometem com uma comunidade cliente alvo (*Target Customer Community*) por meio de uma oferta de serviço (*Service Offering*).

Esta oferta, por sua vez, se caracteriza pela promessa do provedor de serviços (*Service Provider*) de fornecer um serviço, sob certas condições, a uma comunidade cliente alvo (*Target Customer Community*) composta de agentes desempenhando papéis (*Target Customer*) na comunidade (FALBO et al., 2015a). Na Tabela 2 são apresentadas algumas características do padrão e a Figura 12 apresenta o modelo conceitual.

Tabela 2 – Especificação parcial do padrão SOffering (*Service Offering*) (FALBO et al., 2015a) adaptado

Característica	Descrição
Name/Achronym	Organization Provider (O-Provider)
Alternative Patterns	
Intent	Represents organizations as <i>Service Providers</i> .
Rationale	<i>Organizations</i> can act as (play the role of) <i>Service Providers</i> , i.e., the ones responsible for the service provision.
Competency Questions	- Which are the types of service providers?



Figura 12 – Modelo conceitual do padrão SOffering (*Service Offering*) (FALBO et al., 2015a)

3.1.1.3 O-OU-Customer - *Organization/Organizational Unit Customer*

Neste padrão, tem-se a representação das organizações e unidades organizacionais como clientes de serviços. As organizações (*Organization*) são os agentes sociais que têm como objetivo principal a prestação de serviços ou produtos e que são formalmente reconhecidos pelo seu ambiente externo. Já as unidades organizacionais (*Organizational Unit*) são os grupos de trabalho de uma organização (*Organization*) que só são reconhecidos no contexto interno destas organizações (*Organization*). Os serviços prestados pelas organizações (*Organization*) e unidades organizacionais (*Organizational Unit*) são contratados pelos clientes dos serviços (*Service Customer*). Porém, neste padrão, as organizações e unidades organizacionais têm o papel de consumidoras de serviço como unidades organizacionais clientes de serviços (*Organizational Unit Service Customer*) e organizações clientes de serviços (*Organization Service Customer*) (FALBO et al., 2015a). A Tabela 3 são apresentadas algumas características do padrão e a Figura 13 apresenta o modelo conceitual.

Tabela 3 – Especificação parcial do padrão O-OU-Customer (*Organization/Organizational Unit Customer*) (FALBO et al., 2015a) adaptado

Característica	Descrição
Name/Achronym	Organization Provider (O-Provider)
Alternative Patterns	
Intent	Represents organizations as <i>Service Providers</i> .
Rationale	<i>Organizations</i> can act as (play the role of) <i>Service Providers</i> , i.e., the ones responsible for the service provision.
Competency Questions	- Which are the types of service providers?

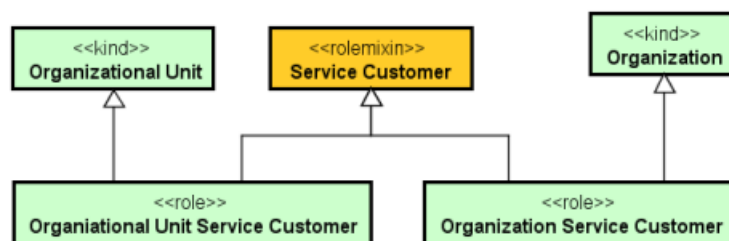


Figura 13 – Modelo conceitual do padrão O-OU-Customer (*Organization/Organizational Unit Customer*) (FALBO et al., 2015a)

3.2 Domínio de Processo de Software

A Engenharia de Software (ES) foca nas diversas fases do processo de construção de uma solução computacional. Entre as atividades realizadas estão a especificação, desenvolvimento, manutenção e criação de sistemas de software que são realizadas com aplicação de tecnologias e práticas de gerência de projetos e outras disciplinas que têm

por objetivo garantir organização, produtividade, qualidade e sistematização. Segundo Falbo et al. (2015b), um dos desafios permanentes em Engenharia de Software é lidar com questões de qualidade, visando entregar um produto de software resultante com maior produtividade e custos mais baixos. Logo, as organizações de software estão investindo cada vez mais na melhoria de seus processos de software.

Segundo Ruy et al. (2015a), neste contexto, vários padrões de qualidade e modelos de maturidade, como ISO/IEC 12207, ISO/IEC 15504, CMMI, entre outros, têm sido utilizados para orientar os esforços das organizações de software para processos de software de qualidade. Essas iniciativas buscam a melhoria do processo de software por meio da disseminação das melhores práticas de forma organizada e padronizada evitando inconsistências devido a utilização diferentes padrões em conjunto, causados por problemas de interoperabilidade semântica. Nesse sentido, foram realizados esforços para criação de uma OPL de processo de software (ISP-OPL) para apoiar a criação de ontologias do domínio de processo de software (FALBO et al., 2015b; RUY et al., 2015a).

Os padrões que compõem a ISP-OPL foram obtidos a partir de uma análise ontológica da ISO/IEC 24744 e, alguns deles, importados e adaptados de SP-OPL e E-OPL, conforme Falbo et al. (2015b), em que são abordados três aspectos principais tratados pelos padrões de processo de software ISO: Unidades de Trabalho (*Work Unit*), incluindo padrões para definir a composição das unidades de trabalho, dependência entre unidades de trabalho e programação de unidades de trabalho, entre outros; Produtos de Trabalho (*Work Product*), considerando a natureza dos produtos de trabalho produzidos durante o processo de software e como eles são manipulados em unidades de trabalho; e Recursos Humanos (*Human Resource*), lidando com a forma como as pessoas são organizadas em equipes, alocadas para tarefas e executar unidades de trabalho.

O interesse desta pesquisa é no conjunto de DROPs definidos para a ISP-OPL (FALBO et al., 2015b; RUY et al., 2015a). O objetivo é estruturar e organizar os DROPs da OPL de processo de software, na forma de catálogo de DROPs, em uma ferramenta computacional (OLED), para apoiar os engenheiros de ontologias na construção de ontologias operacionais de processo de software, servindo, inclusive, de apoio para a própria ISP-OPL, uma vez que com essa ferramenta será possível criar e utilizar catálogos de DROPs que são a base das OPLs.

3.2.1 Padrões Ontológicos de Processo de Software

Os padrões ontológicos de processo de software baseados na ISO ² apresentados a seguir são provenientes da ISP-OPL (FALBO et al., 2015b; RUY et al., 2015a), uma linguagem de padrões ontológicos no domínio de processo de software. Esses padrões foram

² <https://nemo.inf.ufes.br/wp-content/uploads/2015/03/ISP-OPL_Specification_v1.0.pdf>

extraídos a partir de uma reengenharia de fragmentos resultantes da análise ontológica do SEMDM³, bem como da linguagem de padrões ontológicos empresariais (E-OPL) (FALBO et al., 2014). A seguir, apresentamos a definição, parcial, de alguns padrões da ISP-OPL que iremos utilizar para validar a abordagem de catálogo de padrões proposta neste trabalho. A especificação completa pode ser vista no documento de especificação da ISP-OPL (FALBO et al., 2015b).

3.2.1.1 DocD - *Document Depiction*

Este padrão representa os produtos de trabalho (*Work Product*) que um documento (*Document*) descreve. Este padrão pode ser usado para modelar o fato de que documentos retratam (*Document*) outros produtos de trabalho (*Work Product*). A Tabela 4 apresenta algumas características do padrão e a Figura 14 apresenta o modelo conceitual.

Tabela 4 – Especificação parcial do padrão DocD - *Document Depiction* (FALBO et al., 2015b) adaptado*

Característica	Descrição
Name/Achronym	Document Depiction (DocD)
Alternative Patterns	- -
Intent	To represent the work products that a document depicts.
Rationale	A <i>Document</i> has relevant information about an endeavor. It can have independent information, or information about other Work Products <i>Work Product</i> . In the last case, we say that a <i>document</i> depicts <i>work products</i> .
Competency Questions	- Which Work Products does a Document depict?

* Apesar desta dissertação ser escrita em português, as especificações serão mantidas em inglês por ser a língua original da especificação dos padrões

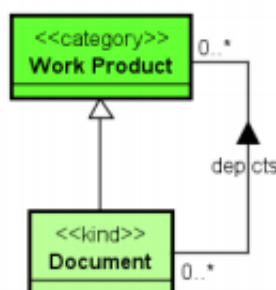


Figura 14 – Modelo conceitual do padrão DocD (*Document Depiction*) (FALBO et al., 2015b)

³ ISO/IEC, ISO/IEC 24744. Software Engineering Metamodel for Development Methodologies, 2007.

3.2.1.2 PPs - *Producer Participation simplified*

Este padrão lida com a participação dos interessados (*Stakeholder*) em unidades de trabalho realizadas (*Performed Work Unit*). O engenheiro de ontologia tem a opção de escolher o padrão alternativo (PPa). A diferença refere-se a se a participação do produtor relacionado (*Producer Participation*) é explicitamente representada ou não. No caso deste padrão, ele não é representado explicitamente. A Tabela 5 apresenta algumas características do padrão e a Figura 15 apresenta o modelo conceitual.

Tabela 5 – Especificação parcial do padrão PPs - *Producer Participation simplified* (FALBO et al., 2015b) adaptado

Característica	Descrição
Name/Achronym	Producer Participation simplified (PPas)
Alternative Patterns	- Producer Participation (PPa)
Intent	To represent the stakeholders participating in a performed work unit.
Rationale	<i>Stakeholders</i> participate in <i>Performed Work Units</i> . In such participations, they act as <i>Producers</i> .
Competency Questions	- Who were the Stakeholders that participated in a Performed Work Unit?

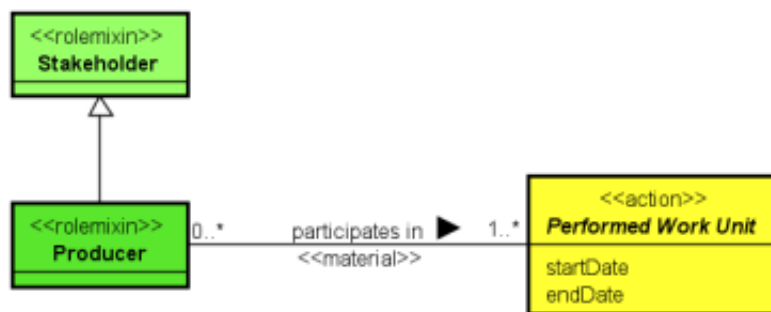


Figura 15 – Modelo conceitual do padrão PPs (*Producer Participation simplified*) (FALBO et al., 2015b)

3.2.1.3 PPP - *Project Process Performing*

Este padrão lida com os processos executados (*Performed Process*) nos contextos de cada projeto (*Project*). Nele podemos ver que vários processos podem ser executados em um projeto. A Tabela 6 apresenta algumas características do padrão e a Figura 16 apresenta o modelo conceitual.

Tabela 6 – Especificação parcial do padrão PPP - *Project Process Performing* (FALBO et al., 2015b) adaptado

Característica	Descrição
Name/Achronym	Project Process Performing (PPP)
Alternative Patterns	- -
Intent	To represent the project in which context a performed process was executed.
Rationale	A <i>Performed Process</i> is a process that was executed in the context of a <i>Project</i> . Many processes can be performed in the context of a project, for different purposes.
Competency Questions	- In the context of which project was a process performed?

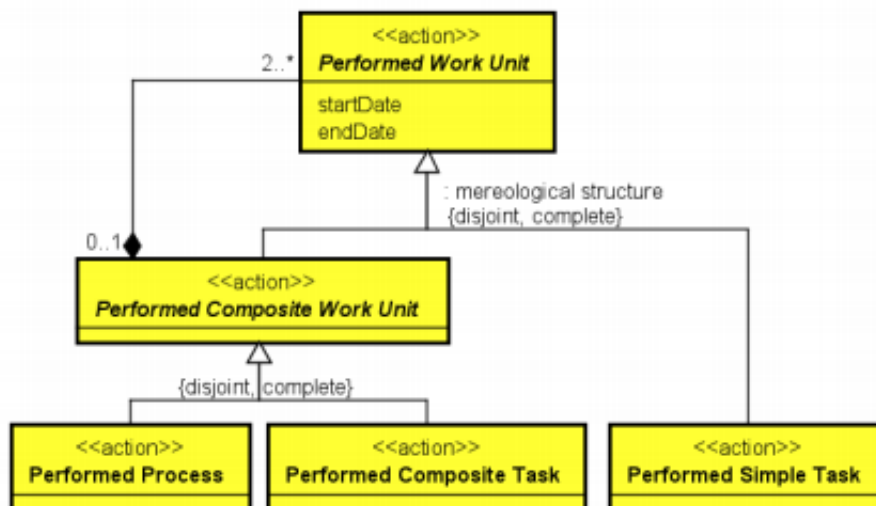
Figura 16 – Modelo conceitual do padrão PPP (*Project Process Performing*) (FALBO et al., 2015b)

3.2.1.4 PWUC - *Performed Work Unit Composition*

Este padrão representa as unidades de trabalho executadas (*Performed Work Unit*) como uma composição de outras unidades de trabalho executadas (*Performed Work Unit*), ou seja, unidades de trabalho executadas (*Performed Work Unit*) podem ser compostas por outras unidades de trabalho executadas (*Performed Work Unit*). Mereologicamente, uma unidade de trabalho executada é simples (*Performed Simple Task*) ou composta por duas ou mais partes (*Performed Composite Work Unit*). No nível básico, são executadas tarefas simples (*Performed Simple Task*) que compõem outras unidades de trabalho executadas (*Performed Work Unit*), mas não são decompostas. Tarefas compostas executadas (*Performed Composite Task*), por sua vez, são compostas por outras tarefas executadas (tarefas compostas ou simples executadas). No nível superior, os processos executados (*Performed Process*) também são compostos de tarefas executadas, mas não compõem outras unidades de trabalho executadas. A Tabela 7 apresenta algumas características do padrão e a Figura 17 apresenta o modelo conceitual.

Tabela 7 – Especificação parcial do padrão PWUC - *Performed Work Unit Composition* (FALBO et al., 2015b) adaptado

Característica	Descrição
Name/Achronym	Performed Work Unit Composition (PWUC)
Alternative Patterns	- -
Intent	To represent the composition of performed work units in terms of other performed work units.
Rationale	<i>Performed Work Units</i> can be composed of other <i>performed work units</i> . Mereologically, a <i>performed work unit</i> is simple, or composed of two or more parts. At the basic level, there are <i>Performed Simple Tasks</i> that compose other <i>performed work units</i> , but are not decomposed. <i>Performed Composite Tasks</i> , in turn, are composed of other performed tasks (<i>composite</i> or <i>simple performed tasks</i>). At the higher level, <i>Performed Processes</i> are also composed of performed tasks, but do not compose other <i>performed work units</i> .
Competency Questions	<ul style="list-style-type: none"> - When a work unit was performed? - Concerning their mereological structure, what are the possible types of performed work units? - How a performed work unit is composed of performed work units?

Figura 17 – Modelo conceitual do padrão PWUC (*Performed Work Unit Composition*) (FALBO et al., 2015b)

3.2.1.5 PWUD - *Performed Work Unit Dependence*

Este padrão representa as unidades de trabalho executadas (*Performed Work Unit*) como dependência de outras unidades de trabalho executadas (*Performed Work Unit*), ou seja, unidades de trabalho executadas (*Performed Work Unit*) podem depender de outras

unidades de trabalho executadas (*Performed Work Unit*). A Tabela 7 apresenta algumas características do padrão e a Figura 17 apresenta o modelo conceitual.

Tabela 8 – Especificação parcial do padrão PWUD - *Performed Work Unit Dependence* (FALBO et al., 2015b) adaptado

Característica	Descrição
Name/Achronym	Performed Work Unit Dependence (PWUD)
Alternative Patterns	- -
Intent	To represent the dependencies between performed work units.
Rationale	For a <i>Performed Work Unit</i> (an actual occurrence of an action) to be accomplished, other <i>performed work units</i> may have to be accomplished first, indicating that there is a dependency between them . There are several situations in which dependencies can occur, such as a <i>performed work unit</i> that has used work products created by another <i>performed work unit</i> , or a <i>performed work unit</i> that has needed a resource which were being used by another <i>performed work unit</i> .
Competency Questions	- When a work unit was performed? - From which work unit did a work unit depend on to be performed?

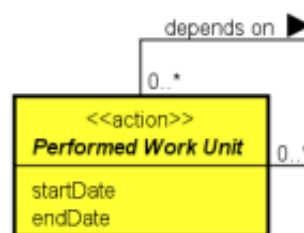


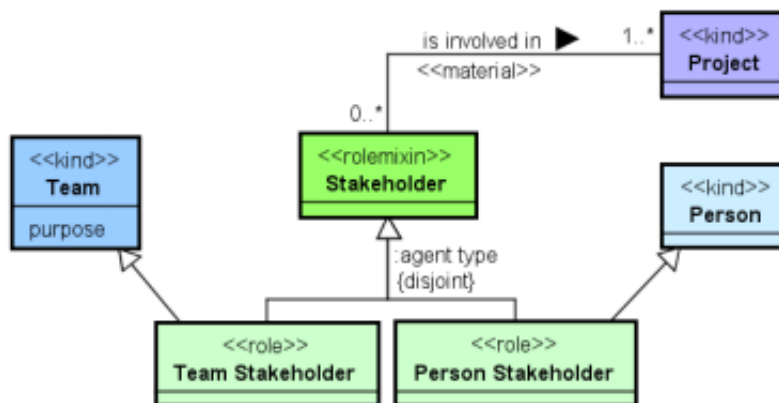
Figura 18 – Modelo conceitual do padrão PWUD (*Performed Work Unit Dependence*) (FALBO et al., 2015b)

3.2.1.6 StD - Stakeholder Definition

Este padrão representa o envolvimento dos interessados (*Stakeholder*) em um projeto (*Project*). Os interessados (*Stakeholder*) pode ser pessoas *Person Stakeholder* ou equipes *Team Stakeholder*. A Tabela 9 apresenta algumas características do padrão e a Figura 19 apresenta o modelo conceitual.

Tabela 9 – Especificação parcial do padrão StD - *Stakeholder Definition* (FALBO et al., 2015b) adaptado

Característica	Descrição
Name/Achronym	Stakeholder Definition (StD)
Alternative Patterns	- -
Intent	To represent the stakeholders (people or teams) involved in projects.
Rationale	<i>Stakeholders</i> are the agents involved in a <i>Project</i> . They can be <i>Team Stakeholders</i> , representing <i>Teams</i> involved in a <i>Project</i> , or <i>Person Stakeholders</i> , representing <i>People</i> involved in a <i>Project</i> .
Competency Questions	- Who are the Stakeholders of a Project? - According to the types of agents, what are the possible types of Stakeholder?

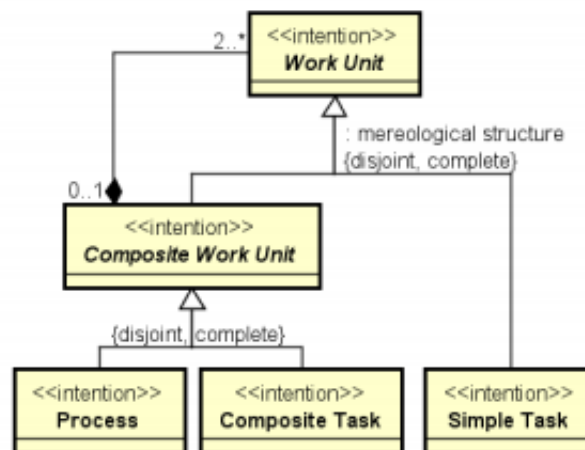
Figura 19 – Modelo conceitual do padrão StD (*Stakeholder Definition*) (FALBO et al., 2015b)

3.2.1.7 WUC - *Work Unit Composition*

Este padrão representa as unidades de trabalho (*Work Unit*) como uma composição de outras unidades de trabalho (*Work Unit*), ou seja, unidades de trabalho (*Work Unit*) podem ser compostas por outras unidades de trabalho (*Work Unit*). Mereologicamente, uma unidade de trabalho é simples (*Simple Task*) ou composta por duas ou mais partes (*Composite Work Unit*). No nível básico, tarefas simples (*Simple Task*) que compõem outras unidades de trabalho (*Work Unit*), mas não são decompostas. Tarefas compostas (*Composite Task*), por sua vez, são compostas por outras tarefas (tarefas compostas ou simples executadas). No nível superior, os processos (*Process*) também são compostos de tarefas, mas não compõem outras unidades de trabalho. A Tabela 10 apresenta algumas características do padrão e a Figura 20 apresenta o modelo conceitual.

Tabela 10 – Especificação parcial do padrão WUC - *Work Unit Composition* (FALBO et al., 2015b) adaptado

Característica	Descrição
Name/Achronym	Work Unit Composition (WUC)
Alternative Patterns	- -
Intent	To represent the composition of work units in terms of other work units.
Rationale	<i>Work Units</i> can be composed of other <i>Work Units</i> . Mereologically, a <i>Work Unit</i> is simple or composed of two or more parts. At the basic level, there are <i>Simple Tasks</i> that compose other <i>Work Units</i> , but are not decomposed. <i>Composite Tasks</i> , in turn, are composed of other tasks (<i>Composite Task</i> or <i>Simple Task</i>). At the higher level, <i>Processes</i> are also composed of tasks, but do not compose other <i>work units</i> .
Competency Questions	<ul style="list-style-type: none"> - Concerning their mereological structure, what are the possible types of work units? - How a work unit is composed of work units?

Figura 20 – Modelo conceitual do padrão WUC (*Work Unit Composition*) (FALBO et al., 2015b)

3.2.1.8 WPChan - *Work Product Change*

Este padrão representa a modificação de produtos de trabalho (*Work Product*) por unidades de trabalho executadas (*Performed Work Unit*), ou seja, unidades de trabalho executadas (*Performed Work Unit*) podem modificar produtos de trabalho (*Work Product*). A Tabela 11 apresenta algumas características do padrão e a Figura 21 apresenta o modelo conceitual.

Tabela 11 – Especificação parcial do padrão WPChan - *Work Product Change* (FALBO et al., 2015b) adaptado

Característica	Descrição
Name/Achronym	Work Product Change (WPChan)
Alternative Patterns	- -
Intent	To represent the change of work products during the execution of performed work units.
Rationale	<i>Performed Work Units</i> can change <i>Work Products</i> .
Competency Questions	- What are the Work Products changed by a given Performed Work Units?

Figura 21 – Modelo conceitual do padrão WPChan (*Work Product Change*) (FALBO et al., 2015b)

3.2.1.9 WPCrea - *Work Product Creation*

Este padrão representa a criação de produtos de trabalho (*Work Product*) por unidades de trabalho executadas (*Performed Work Unit*), ou seja, unidades de trabalho executadas (*Performed Work Unit*) podem criar produtos de trabalho (*Work Product*). A Tabela 12 apresenta algumas características do padrão e a Figura 22 apresenta o modelo conceitual.

Tabela 12 – Especificação parcial do padrão WPCrea - *Work Product Creation* (FALBO et al., 2015b) adaptado

Característica	Descrição
Name/Achronym	Work Product Creation (WPCrea)
Alternative Patterns	- -
Intent	To represent the creation of work products by performed work units.
Rationale	<i>Performed Work Units</i> can create <i>Work Products</i> during their execution.
Competency Questions	- What are the Work Products created by a Performed Work Unit?

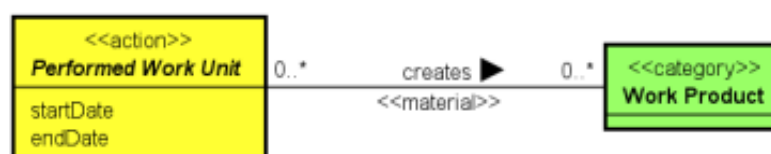


Figura 22 – Modelo conceitual do padrão WPCrea (*Work Product Creation*) (FALBO et al., 2015b)

3.2.1.10 WPUse - *Work Product Usage*

Este padrão representa o uso de produtos de trabalho (*Work Product*) por unidades de trabalho executadas (*Performed Work Unit*), ou seja, unidades de trabalho executadas (*Performed Work Unit*) podem usar produtos de trabalho (*Work Product*). A Tabela 13 apresenta algumas características do padrão e a Figura 23 apresenta o modelo conceitual.

Tabela 13 – Especificação parcial do padrão WPUse - *Work Product Usage* (FALBO et al., 2015b) adaptado

Característica	Descrição
Name/Achronym	Work Product Usage (WPUse)
Alternative Patterns	- -
Intent	To represent the usage of work products by performed work units.
Rationale	<i>Performed Work Units</i> can use <i>Work Products</i> .
Competency Questions	- What are the Work Products used by a given Performed Work Unit?



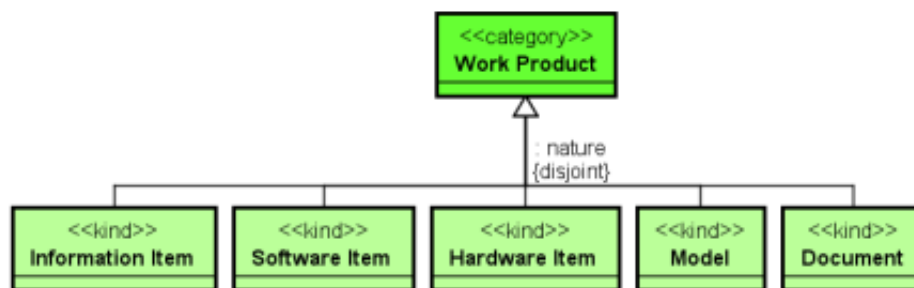
Figura 23 – Modelo conceitual do padrão WPUse (*Work Product Usage*) (FALBO et al., 2015b)

3.2.1.11 WPN - *Work Product Nature*

Este padrão representa os diferentes tipos produtos de trabalho (*Work Product*) conforme a natureza deles. As naturezas mais comuns dos produtos de trabalho e que foram mapeadas neste OP são documento (*Document*), modelo (*Model*), item de *software* (*Software Item*), item de informação (*Information Item*) e item de *hardware* (*Hardware Item*). A Tabela 14 apresenta algumas características do padrão e a Figura 24 apresenta o modelo conceitual.

Tabela 14 – Especificação parcial do padrão WPN - *Work Product Nature* (FALBO et al., 2015b) adaptado

Característica	Descrição
Name/Achronym	Work Product Nature (WPN)
Alternative Patterns	- -
Intent	To represent the different types of work products, according to their nature.
Rationale	<i>Work Products</i> can have different natures. The most common are <i>Document</i> , <i>Model</i> , <i>Software Item</i> , <i>Information Item</i> , and <i>Hardware Item</i> .
Competency Questions	- According to its nature, what is the type of a Work Product?

Figura 24 – Modelo conceitual do padrão WPN (*Work Product Nature*) (FALBO et al., 2015b)

3.3 Domínio de Colaboração

A colaboração faz parte dos processos do dia-a-dia, apoiando as atividades por meio das interações realizadas entre os envolvidos. Webster (2016) define colaboração como trabalhar com outra pessoa ou grupo a fim de alcançar ou fazer algo. Ou seja, sempre que houver contribuição de outrem na execução de alguma atividade, isso se dará pela colaboração. Mas vale ressaltar que a colaboração não se restringe apenas a pessoas, ela pode se dar entre outros agentes como sistemas computacionais, animais, organizações, serviços, etc como nos exemplos de um caixa eletrônico bancário que executa uma tarefa para o cliente do banco, um jôquei e seu cavalo que correm em busca da vitória em uma prova de corrida, sistemas de diferentes empresas e órgãos governamentais interagindo a fim de gerar informação. A colaboração eficaz é um fator determinante para se alcançar os objetivos e, por isso, deve se utilizar todos os recursos que estiverem disponíveis.

No contexto dos negócios, não é diferente. Conforme Oliveira (2009a), a colaboração é de extrema importância para a tomada de decisões. Uma organização que colabora de forma eficiente pode maximizar seus lucros através da redução dos custos de transação entre as organizações e permitindo a criação de modelos de negócios especializados, combinando

serviços prestados por outras organizações. A colaboração é ferramenta essencial uma vez que é usada para trocar informações internamente, dentro de uma organização, e também externamente, entre organizações (TERRA et al., 2002).

Um grande desafio que surge nesse contexto diz respeito a integração. Num contexto de integração, com sistemas desenvolvidos isoladamente, inevitavelmente geram incompatibilidades pois foram concebidos por diferentes visões de um mesmo problema e muito provavelmente foram modelados utilizando-se linguagens e padrões diferentes uns dos outros. Isso produz incompatibilidades, haja visto que, cada ferramenta tem sua própria semântica e, portanto, a unificação da representação de cada uma delas através de um mapeamento em uma semântica compartilhada se torna necessário para que uma possa entender a outra (OLIVEIRA, 2009a).

Aqui, análogo às seções anteriores, o interesse da nossa pesquisa se refere a uma coleção de DROPs. Nesta seção, focalizaremos os padrões definidos com base na Ontologia de Colaboração (CONTO) (OLIVEIRA, 2009a; SOUZA et al., 2016). É importante ressaltar que, diferentemente dos padrões descritos em 3.1 e 3.2, os padrões aqui descritos não foram ainda publicados em artigos ou outra referência bibliográfica, a não ser por um ou dois exemplos utilizados como ilustração em (SOUZA et al., 2016) sem ser, entretanto, o foco principal de tal artigo. Assim, o objetivo desta seção é apresentar, de maneira estruturada, a especificação dos DROPs baseados na CONTO. Ressalta-se, ainda, que tais padrões se encontram, ainda, em fase de construção.

3.3.1 Ontologia de Colaboração (CONTO)

Os critérios utilizados para estruturar a ontologia de colaboração (CONTO) são aqueles propostos no modelo 3C, que é amplamente utilizado em várias soluções referentes à colaboração como um meio de classificar sistemas colaborativos (BORGHOFF; SCHLICHTER, 2013) e como um meio para criar um processo de desenvolvimento compartilhado (*groupware*) (PIMENTEL, 2006). A Figura 26 apresenta as três subontologias de CONTO, seguindo a estrutura do modelo 3C: cooperação, comunicação e coordenação. Estas três ontologias representam o conhecimento capturado do domínio específico de colaboração com os respectivos conceitos, relacionamentos, propriedades de domínio e um conjunto de axiomas que restringem o domínio colaboração. A Figura 26 ilustra essas ontologias a partir de um diagrama de pacote UML, modelando as três ontologias e as relações entre elas.

A Ontologia de Cooperação tem por objetivo apresentar as relações entre as entidades na realização de atividades em conjunto, ou seja, na cooperação entre elas, interagindo para a concretização de uma tarefa. As atividades e tarefas conjuntas, bem como o local e os recursos disponíveis são de fundamental importância para a caracterização da cooperação. Assim, o problema com que esta ontologia lida está relacionado ao contexto

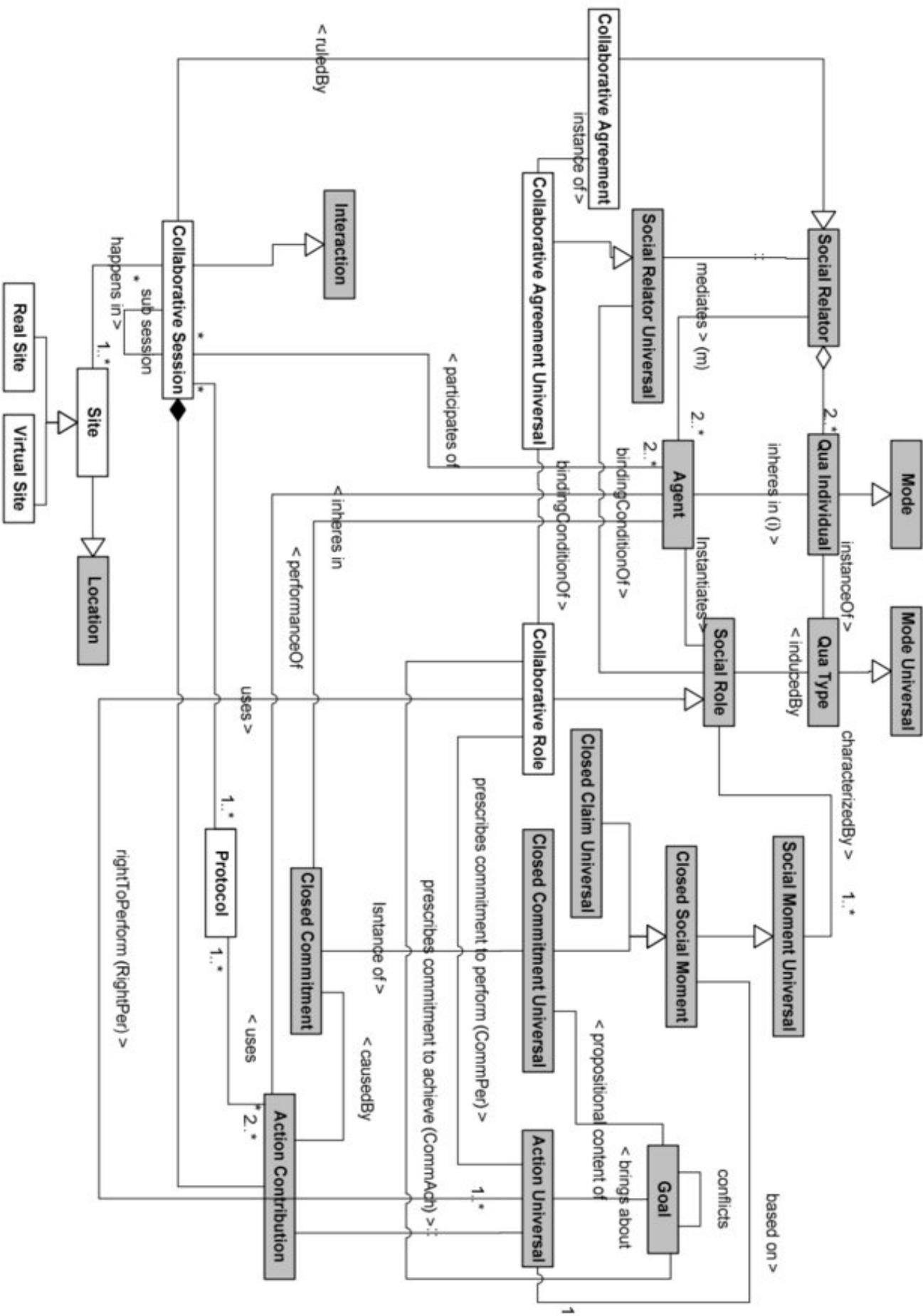


Figura 25 – Ontologia de cooperação (OLIVEIRA, 2009a)

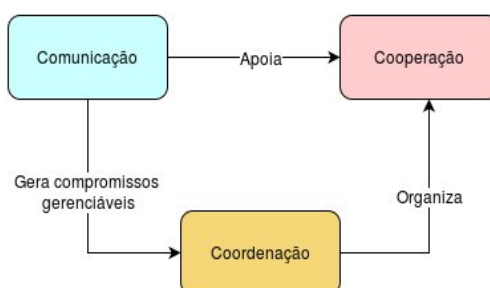


Figura 26 – As três subontologias que compõem a ontologia de colaboração (CONTO)

em que a cooperação acontece e as relações que as entidades têm neste contexto. A Figura 25 apresenta a ontologia de referência de cooperação e a Tabela 15, a definição dos termos.

Tabela 15 – Dicionário de termos da ontologia de cooperação (OLIVEIRA, 2009a)

Termo	Descrição
Collaborative Session	Eventos complexos nos quais participantes interagem através de participações/contribuições com o propósito de colaborar.
Site	Local onde a sessão colaborativa acontece.
Site Virtual	Ambiente que é simulado por uma ferramenta de software onde uma sessão colaborativa acontece.
Site Real	Ambiente concreto do mundo real onde a sessão colaborativa acontece.
Collaborative Agreement	Representa um tipo de papel social que rege uma sessão colaborativa.
Collaborative Agreement Universal	Tipo de Social Relator Universal que pode ser instanciado por um Collaborative Agreement e fornece a condição de ligação de um Papel colaborativo.
Collaborative Role	Papel Social que agentes se comprometem a desempenharem devido aos acordos por eles assumidos.

Já a Ontologia de Comunicação tem por objetivo apresentar as relações entre as entidades na realização do processo de comunicação, ou seja, durante o processo em que as entidades estão enviando e recebendo mensagens umas das outras para a concretização de uma tarefa. As mensagens, o idioma, bem como quem recebe e envia as mesmas é de fundamental importância para a caracterização da comunicação. A Figura 27 traz a ontologia de referência de comunicação e a Tabela 16, a definição dos termos.

A Ontologia de Coordenação tem por objetivo apresentar as relações entre as entidades na realização da organização e harmonização para os conceitos presentes nas ontologias de Cooperação e Comunicação, ou seja, ela tem como objetivo organizar e harmonizar a colaboração como um todo, capturando, as relações de dependência entre

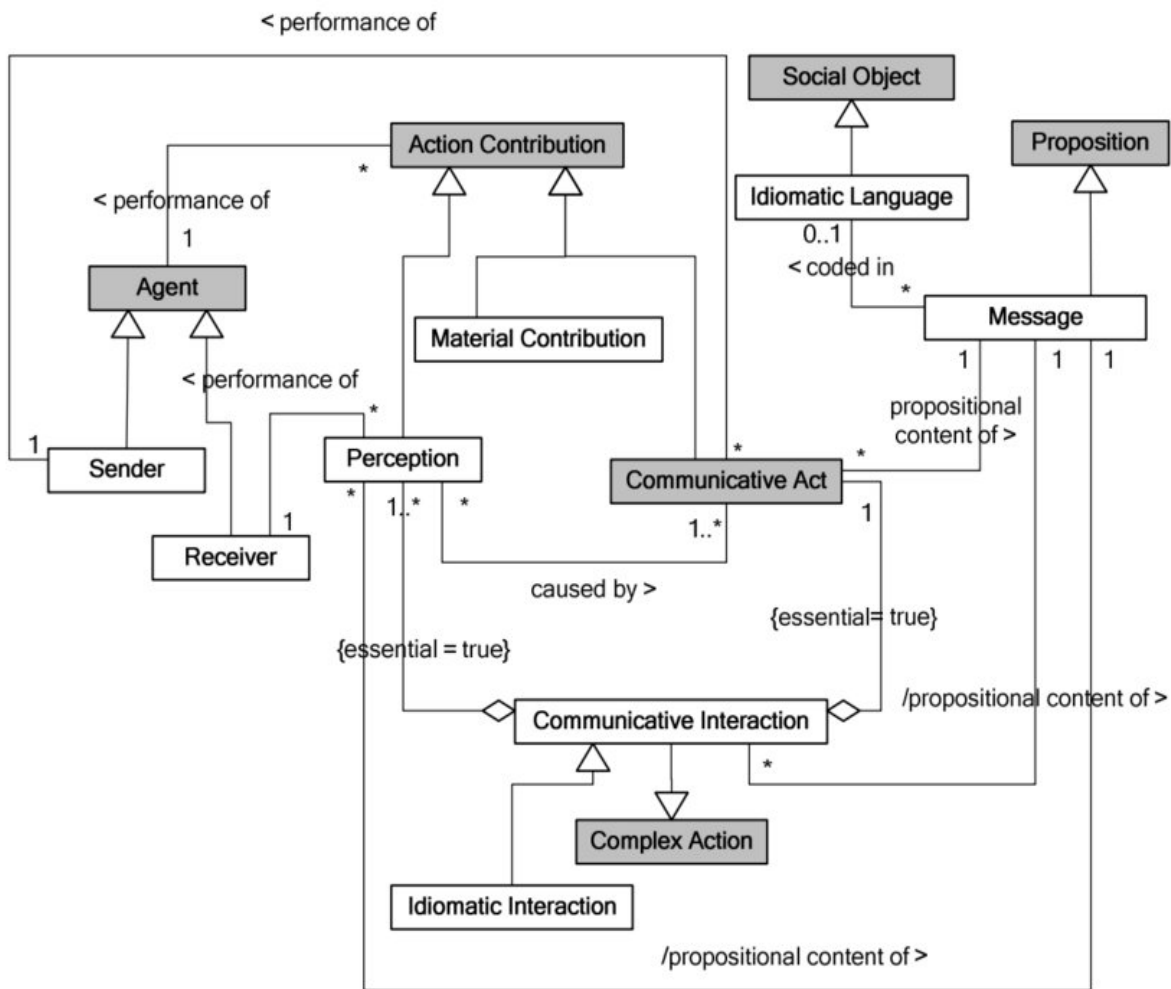


Figura 27 – Ontologia de comunicação (OLIVEIRA, 2009a)

Tabela 16 – Dicionário de termos da ontologia de comunicação (OLIVEIRA, 2009a)

Termo	Descrição
Communicative Interaction	Representa a comunicação entre dois ou mais agentes através de troca de informações: envio e percepção.
Material Contribution	Ação de Contribuição em que não ocorre troca de mensagens.
Message	A mensagem é o conteúdo proposicional trocado de um ato de comunicação e, conseqüentemente, de uma interação comunicativa entre agentes.
Communicative Act	Representa a ato de enviar uma mensagem.
Perception	Representa a ação de perceber uma mensagem.
Idiomatic Language	Linguagem que usa um idioma para a sua representação.
Sender	Indivíduo capaz de emitir uma mensagem.
Receiver	Indivíduo capaz de perceber uma mensagem.
Idiomatic Interaction	Interação Comunicativa que envolve uma mensagem codificada em uma linguagem idiomática.

ações potenciais de colaboração e possíveis conflitos, dado que uma premissa para que a colaboração aconteça entre indivíduos é que os mesmos não assumam objetivos conflitantes no contexto da colaboração. Nas Figura 28 e 29, encontram-se fragmentos da ontologia de referência de comunicação e na Tabela 17, a definição dos termos.

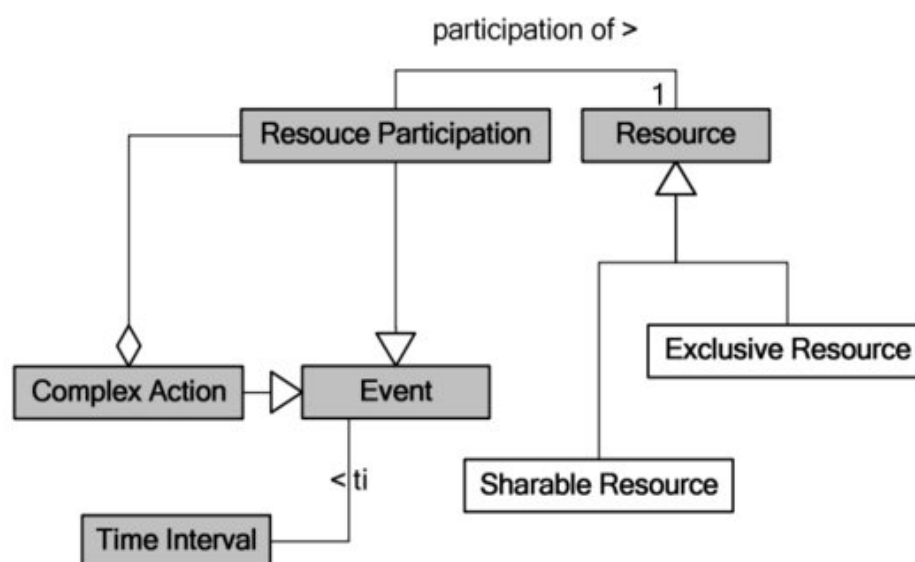


Figura 28 – Fragmento da ontologia de coordenação (OLIVEIRA, 2009a)

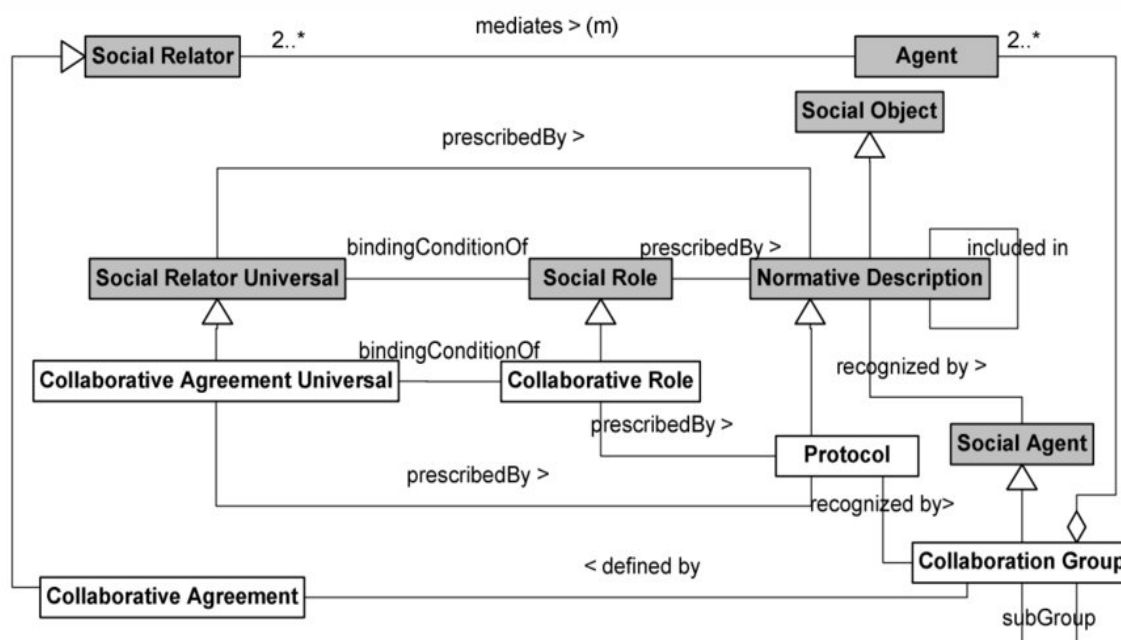


Figura 29 – Fragmento da ontologia de coordenação representando conceitos de grupos e protocolos e as respectivas relações (OLIVEIRA, 2009a)

Tabela 17 – Dicionário de termos da ontologia de coordenação (OLIVEIRA, 2009a)

Termo	Descrição
Protocol	Conjunto de regras que estabelece coordenação para a harmonia e melhoria da Sessão Colaborativa.
Collaborative Group	Conjunto de agentes definidos por Acordos Colaborativos.
Exclusive Resource	Recurso que não pode ser utilizado de forma simultânea.
Sharable Resource	Recurso que pode ser utilizado de forma simultânea.
Flow	Dependência entre ações em que uma delas cria um recurso que é requisitado pela outra.
Usability	Especialização de uma dependência do tipo Flow na qual o recurso produzido deve possuir também características específicas.
Transfer	Especialização de uma dependência do tipo Flow na qual o recurso produzido deve ser disponibilizado em uma localização específica quando.
Fit	Indivíduo capaz de perceber uma mensagem.
Idiomatic Interaction	Dependência de dois ou mais eventos necessitam serem executados simultânea para a criação de um recurso.

3.3.2 Padrões Ontológicos de Colaboração

Os OPs de colaboração foram extraídos diretamente da CONTO. Cada um deles consiste em um conjunto de conceitos que, juntos, modelam uma solução para resolver um problema recorrente de modelagem de ontologia no domínio de colaboração. Eles foram definidos e organizados em um documento de especificação (vide apêndice A) que foi utilizado para escolha dos OPs apresentados nessa seção.

Os padrões mostrados na Figura 30 são: (i) o padrão *Collaborative Session Resource* (CSR), cujo objetivo é identificar quais recursos participam na sessão colaborativa e (ii) o padrão de *Resource Nature* (RN) que permite consultar à ontologia sobre os recursos compartilháveis e exclusivos que participam na sessão colaborativa. As Tabelas ?? e 25 respectivamente, apresentam parte da especificação desses dois OPs.

Dizemos que a especificação é parcial porque no documento de especificação, além das informações apresentadas nessas tabelas, para cada padrão existe um diagrama modelado usando OntoUML e uma axiomatização formalizando o padrão. A definição dos mesmos encontra-se em desenvolvimento e, portanto, são passíveis de modificações até que sejam considerados mais estáveis. A seguir, são apresentados os DROPs definidos para a ontologia de Cooperação, Comunicação e Coordenação. Mais detalhes dos DROPs de colaboração são descritos no capítulo 4.

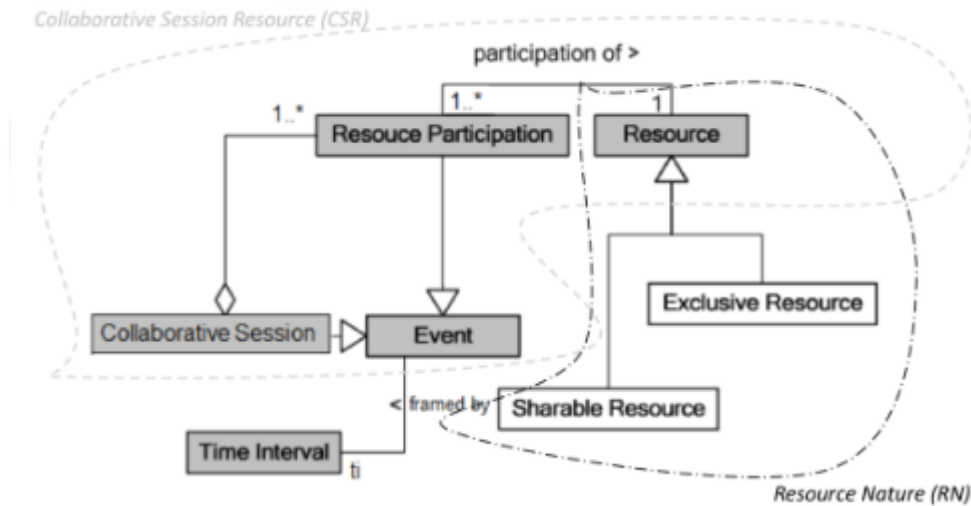


Figura 30 – Extração de OPs de um fragmento da ontologia de cooperação

3.3.2.1 Padrões de Cooperação

3.3.2.1.1 CSL – Collaborative Session Location

Este padrão tem por objetivo explicitar o local (*Site*) onde ocorre a sessão colaborativa (*Collaborative Session*). A Tabela 18 apresenta algumas características do padrão e a Figura 31 apresenta o modelo conceitual.

Tabela 18 – Especificação parcial do padrão *Collaborative Session Location* (CSL)*

Característica	Descrição
Name/Achronym	Collaborative Session Location (CSL)
Alternative Patterns	- -
Intent	To represent the site where the collaborative session happens.
Rationale	A Collaborative Session happens in a specific site, which is a location. This pattern makes this site explicit.
Competency Questions	- Where is the collaborative session?

* Apesar desta dissertação ser escrita em português, as especificações serão mantidas em inglês por ser a língua original da especificação dos padrões

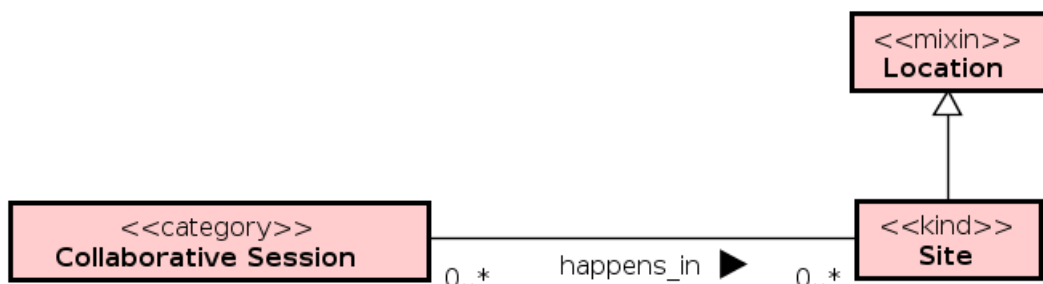


Figura 31 – Collaborative Session Location

3.3.2.1.2 CSAC – Collaborative Session Action Contributions

Este padrão representa as ações de contribuição (*Action Contribution*) que compõem a sessão colaborativa (*Collaborative Session*), ou seja, duas ou mais ações de contribuição (*Action Contribution*) compõem uma sessão colaborativa (*Collaborative Session*). A Tabela 19 apresenta algumas características do padrão e a Figura 32 apresenta o modelo conceitual.

Tabela 19 – Especificação parcial do padrão *Collaborative Session Action Contributions* (CSAC)

Característica	Descrição
Name/Achronym	Collaborative Session Action Contributions (CSAC)
Alternative Patterns	- -
Intent	To represent the action contributions that compose the collaborative session.
Rationale	Two or more action contributions compose a collaborative session.
Competency Questions	- What are the action contributions that compose the collaborative session?

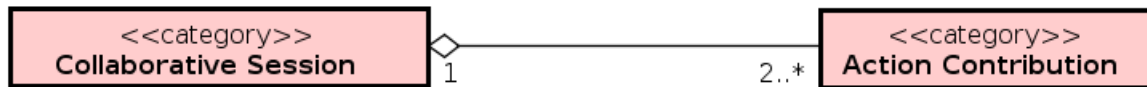


Figura 32 – Collaborative Session Action Contributions

3.3.2.1.3 CSA – Collaborative Session Agents

Este padrão representa os agentes (*Agent*) que participam em uma sessão colaborativa (*Collaborative Session*), ou seja, uma sessão colaborativa (*Collaborative Session*) só existe se houver dois ou mais agentes (*Agent*) participando em uma sessão colaborativa (*Collaborative Session*). A Tabela 20 apresenta algumas características do padrão e a Figura 33 apresenta o modelo conceitual.

Tabela 20 – Especificação parcial do padrão *Collaborative Session Agents* (CSA)

Característica	Descrição
Name/Achronym	Collaborative Session Agents (CSA)
Alternative Patterns	- -
Intent	To represent the agents that participate in the collaborative session.
Rationale	Two or more agents participate in a collaborative session.
Competency Questions	- What are the agents that participate in the collaborative session?



Figura 33 – Collaborative Session Agents

3.3.2.1.4 CCAC – Collaborative Commitment for Action Contributions

Este padrão representa o compromisso colaborativo (*Collaborative Commitment*) que causam ações de colaboração (*Action Contribution*), ou seja, uma ação de colaboração (*Action Contribution*) é causada por um compromisso colaborativo (*Collaborative Commitment*). A Tabela 21 apresenta algumas características do padrão e a Figura 34 apresenta o modelo conceitual.

Tabela 21 – Especificação parcial do padrão *Collaborative Commitment for Action Contributions* (CCAC)

Característica	Descrição
Name/Achronym	Collaborative Commitment for Action Contributions (CCAC)
Alternative Patterns	- -
Intent	To represent the collaborative commitment that causes the action contribution.
Rationale	One action contributions is caused by a collaborative commitment.
Competency Questions	- What is the collaborative commitment that causes the action contribution?

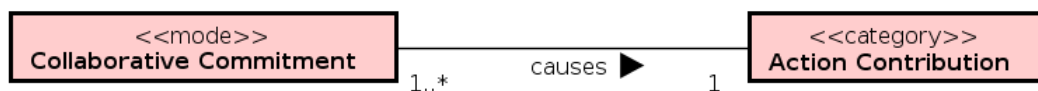


Figura 34 – Collaborative Commitment for Action Contributions

3.3.2.1.5 CCIC – Collaborative Commitment Inheres in Collaborator

Este padrão representa o compromisso colaborativo (*Collaborative Commitment*) que é inerente a um colaborador (*Collaborator*). . A Tabela 22 apresenta algumas características do padrão e a Figura 35 apresenta o modelo conceitual.

Tabela 22 – Especificação parcial do padrão *Collaborative Commitment Inheres in Collaborator* (CCIC)

Característica	Descrição
Name/Achronym	Collaborative Commitment Inheres in Collaborator (CCIC)
Alternative Patterns	- -
Intent	To represent the collaborative commitment that inheres in a collaborator.
Rationale	One collaborative commitment inheres in an collaborator.
Competency Questions	- What is the collaborator that inheres in a collaborative commitment?

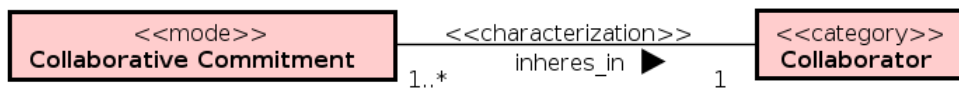


Figura 35 – Collaborative Commitment Inheres in Collaborator

3.3.2.1.6 ACPPS – Action Contribution Pre and Pos Situation

Este padrão representa a situação (*Situation*) antes e depois de uma ação de contribuição (*Action Contribution*). A Tabela 23 apresenta algumas características do padrão e a Figura 36 apresenta o modelo conceitual.

Tabela 23 – Especificação parcial do padrão *Action Contribution Pre and Pos Situation* (ACPPS)

Característica	Descrição
Name/Achronym	Action Contribution Pre and Pos Situation (ACPPS)
Alternative Patterns	- -
Intent	To represent the pre and pos situation of the action contributions.
Rationale	A pre and pos situation of the action contributions.
Competency Questions	- What is the situation before the contribution action? - What is the situation after the contribution action?

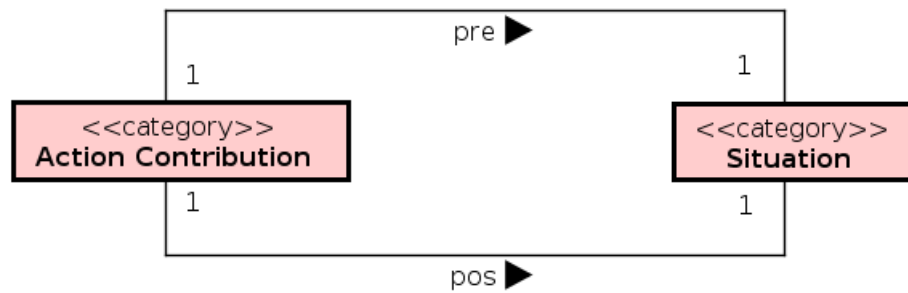


Figura 36 – Action Contribution Pre and Pos Situation

3.3.2.2 Padrões de Coordenação

3.3.2.2.1 CSRS – Collaborative Session Resources Simple

Este padrão representa a participação de recursos (*Resource*) em sessão colaborativa (*Collaborative Session*) (sem a mediação de um agente de participação de recursos (*Resource Participation*)). Um ou mais recursos (*Resource*) participam em uma sessão colaborativa (*Collaborative Session*). A Tabela 24 apresenta algumas características do padrão e a Figura 37 apresenta o modelo conceitual.

Tabela 24 – Especificação parcial do padrão *Collaborative Session Resources Simple* (CSRS)

Característica	Descrição
Name/Achronym	Collaborative Session Resources Simple (CSRS)
Alternative Patterns	- Collaborative Session Resource (CSR)
Intent	To represent the resources that participate in the collaborative session.
Rationale	One or more resources participate in a collaborative session.
Competency Questions	- What are the resources that participate in the collaborative session?



Figura 37 – Collaborative Session Resources Simple

3.3.2.2.2 RN – Resource Nature

Este padrão representa os tipos de recurso (*Resource*) em termos de compartilhamento. Portanto um recurso pode ser compartilhado (*Sharable Resource*) ou exclusivo (*Exclusive Resource*). A Tabela 25 apresenta algumas características do padrão e a Figura

38 apresenta o modelo conceitual.

Tabela 25 – Especificação parcial do padrão *Resource Nature* (RN)

Característica	Descrição
Name/Achronym	Resource Nature (RN)
Alternative Patterns	- -
Intent	To represent resource types in terms of shareability.
Rationale	A resource can be either sharable or exclusive.
Competency Questions	- Can this resource be shared?

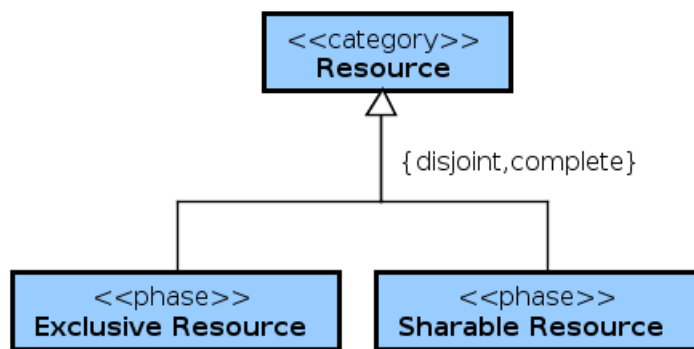


Figura 38 – Resource Nature

4 Abordagem Metodológica para a Criação de Catálogos de DROPs

Este capítulo se destina à proposta de uma abordagem metodológica para uso de catálogos de DROPs no desenvolvimento de ontologias. Além disso, a seção 4.5 deste capítulo propõe uma OPL para os padrões de Colaboração apresentados no capítulo 3. Este capítulo inaugura, assim, a parte núcleo desta dissertação, composta pelos capítulos 4 e 5.

Como já mencionado em capítulos anteriores, a utilização da abordagem de catálogo de DROPs tem por objetivo final fornecer um conjunto de DROPs pertencentes a um domínio específico para apoiar os engenheiros de ontologias na criação de novas ontologias. Assim sendo, o trabalho de identificação e extração dos DROPs a partir de uma ontologia de referência deve ser previamente realizado pelos desenvolvedores do próprio catálogo de DROPs.

Espera-se que, com a leitura deste capítulo, o leitor possa compreender melhor como desenvolver padrões a partir de ontologias de referência. Dessa forma, ao invés de começar do zero, o desenvolvedor de padrões tem, ao seu dispor, um processo, bem como algumas orientações sobre como definir DROPs que possam ser reutilizados no futuro no desenvolvimento de ontologias de domínio. É importante ressaltar que a qualidade dos padrões pode impactar sobremaneira a qualidade das ontologias criadas a partir deles (BLOMQUIST; GANGEMI; PRESUTTI, 2009) e esta dissertação assume que o processo de criação desses padrões pode ter impacto em sua criação.

A Figura 39 ilustra o passo-a-passo aqui proposto para o desenvolvimento de catálogos de DROPs, representando-o a partir de um modelo de processos.

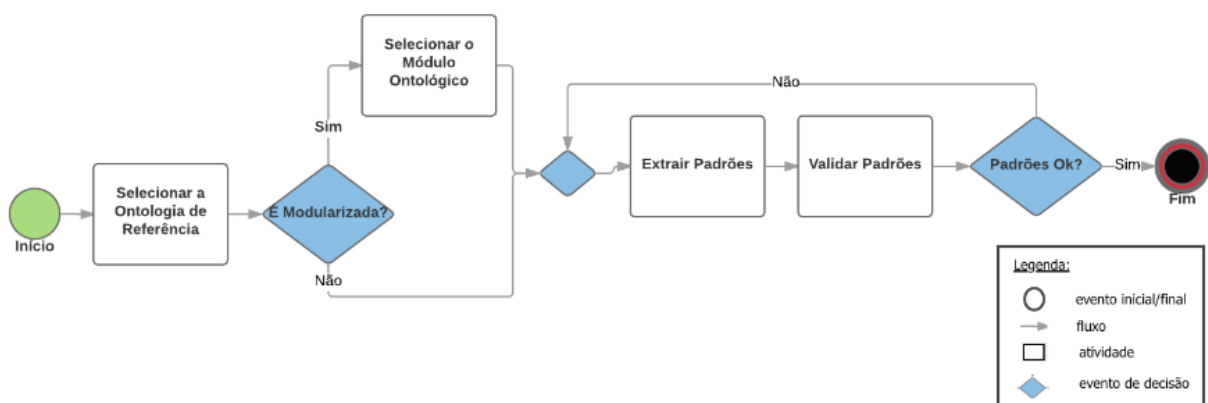


Figura 39 – Processo da abordagem metodológica proposta

O processo da Figura 39 permite que o leitor compreenda qual a sequência de ativi-

dades que compõe a abordagem metodológica. Em primeiro lugar, é preciso **Selecionar a Ontologia de Referência**, que será utilizada como base para a construção dos padrões, conforme já mencionado na seção 2.4. Em segundo lugar, caso a ontologia selecionada seja composta de vários módulos (ou subontologias), é preciso **Selecionar o Módulo Ontológico** específico que se deseja utilizar. Em seguida, já é possível **Extrair Padrões** desse módulo, ao subdividi-lo em pequenos pedaços. Nesse momento, além de extraído, cada padrão deve ser também especificado. Assim que o conjunto de padrões tenha sido definido, os padrões precisam ser validados (atividade **Validar Padrões**), verificando-se sua qualidade e adequação na produção de fragmentos ontológicos. Se o desenvolvedor de padrões concluir que eles têm boa qualidade, o processo termina. Senão, retorna ao ponto em que os padrões devem ser extraídos e validados. As seções 4.1 a 4.4. descrevem cada uma dessas atividades da abordagem em detalhe.

4.1 Selecionar a Ontologia de Referência

A criação de um catálogo de DROPs começa com a escolha de uma ontologia de referência do domínio de onde são extraídos os DROPs que serão organizados/empacotados como um catálogo para que possam ser utilizados na construção de ontologias operacionais em seus domínios originais.

Nesta etapa, é necessário que se leve em consideração em que domínio os padrões serão criados. Assim, se o desenvolvedor quer prover um catálogo para dar suporte à criação de ontologias na área de Serviço, por exemplo, uma ontologia de serviço deve ser selecionada. Além do domínio em questão, deve-se observar o escopo dessa ontologia, normalmente indicado por questões de competência (FALBO, 2011), i.e. perguntas em linguagem natural que a ontologia deve responder. Por fim, medidas de qualidade, tais como para valorar a consistência e completude da ontologia em relação ao domínio (GUIZZARDI, 2005) podem ser úteis para auxiliar essa atividade.

4.2 Selecionar o Módulo Ontológico

Se a ontologia de referência selecionada no passo anterior for estruturada em módulos ou subontologias, então um desses módulos deve ser selecionado para a criação dos padrões. Novamente, o escopo é essencial nesta atividade, e vai determinar do módulo. Em outras palavras, o módulo é selecionado conforme o que se pretende modelar com os padrões a serem criados.

Como exemplo, considere a ontologia desenvolvida para o domínio de Colaboração, chamada CONTO (OLIVEIRA, 2009a) e citada na seção 3.3 desta dissertação. Essa ontologia foi estruturada conforme o modelo 3C o que originou subontologias em três

subáreas: Cooperação, Comunicação e Coordenação. Essas três subontologias representam o conhecimento capturado do domínio específico de colaboração com os respectivos conceitos, relacionamentos, propriedades de domínio e um conjunto de axiomas que restringem o domínio colaboração.

Supondo que o desenvolvedor esteja interessado em prover um catálogo de padrões para subsidiar a criação de ontologias na área de Cooperação, o módulo de CONTO relacionado a essa área será selecionado. A subontologia de Cooperação da CONTO é apresentada na Figura 25, no capítulo 3.

4.3 Extrair Padrões

O módulo ontológico selecionado na atividade anterior deve ser subdividido em pequenos pedaços, cada um objetivando resolver um problema de modelagem. Para facilitar a explicação das orientações metodológicas relacionadas a esta atividade, considere o subontologia de Cooperação da CONTO.

A Figura 40 ilustra a subdivisão da subontologia em pequenos pedaços, identificados por pacotes UML e coloridos para facilitar a identificação e diferenciação entre os padrões. Esses pacotes possuem pequenos identificadores nos cantos superiores esquerdos que são os nomes dados aos DROPs. Nessa ontologia, tem-se DROPs que lidam com acordos, recursos, ações, compromissos e objetivos de colaboração bem como onde todos são realizados. Os DROPs da Figura 40 são os mesmos apresentados na seção 3.3.2.

Cada padrão extraído apresenta uma solução para resolver um problema de modelagem de ontologia recorrente. Por exemplo, o padrão LT, representado em marrom na parte inferior direita da Figura 40 trata do tipo do local da colaboração (i.e. se esse local é real ou virtual). Em várias situações esse problema de modelagem se repete. Tome, como exemplo, um curso parcialmente realizado presencialmente e parcialmente, a distância. Nesse contexto, talvez seja importante modelar que seções colaborativas ocorrem em um local real (ex: uma sala da Universidade) ou em um local virtual (ex.: um website ou uma sala de chat).

Selecionar um DROP não é uma tarefa simples porque, na ontologia de referência, todos os conceitos estão conectados através de relações (veja a Figura 40). E ao extrair um padrão, deve-se ter ciência da granularidade correta, ou seja, quantos conceitos devem fazer parte do padrão. Se o padrão for muito grande, podem haver casos em que alguns dos conceitos serão ignorados quando o padrão for usado na criação de ontologia.

Dividir padrões e ignorar conceitos é indesejável, caso contrário, a reutilização pode ser comprometida e a ontologia resultante pode tornar-se inconsistente. Por outro lado, não faz sentido um padrão que seja composto de um só conceito e sem nenhuma relação.

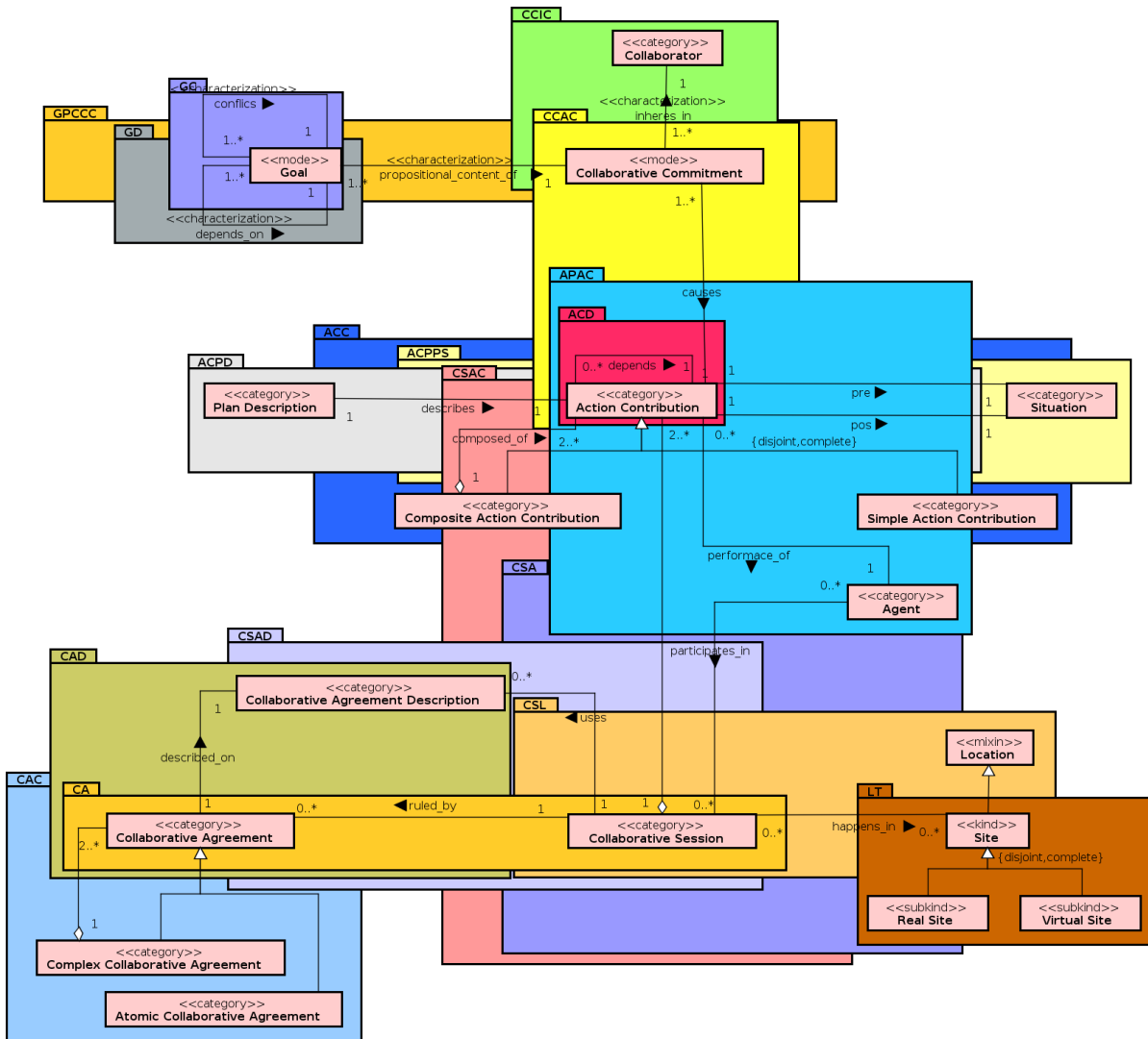


Figura 40 – Ontologia de referência de cooperação da CONTO com os padrões destacados

É difícil oferecer uma regra para a extração de padrões, mas pode-se afirmar que a maior parte dos padrões é composta de DROPs contendo dois conceitos relacionados. Há também alguns que contêm apenas um conceito autorreferenciado e alguns outros compostos por três conceitos. DROPs contendo mais de três conceitos são raros, apesar de possíveis. Se o leitor quiser confirmar tal afirmação, basta examinar os padrões exemplificados no capítulo 3.

Durante a extração, além de definir um padrão para cada problema de modelagem, cada padrão deve ser especificado, documentando-se:

- O nome do padrão;
- O nome de padrões alternativos, se houver;
- Sua intenção (*intention*) - o propósito do padrão;

- Sua justificativa (rationale) - o raciocínio subjacente ao padrão. Uma breve declaração respondendo a seguinte pergunta: qual a justificativa do padrão?;
- As questões de competência a que o padrão deve responder;
- Os axiomas que formalizam o padrão.

Como exemplo de um DROP, na Tabela 26, pode-se ver as especificações do DROP que lida com a natureza de um recurso, pertencente ao domínio de colaboração. Nela, temos algumas características necessárias para definição do DROP: nome, intenção, justificativa, as questões de competência e os axiomas que descrevem o padrão apresentado. Na Figura 41, temos o modelo do DROP RN, em cinza, que é descrito na Tabela 26, sendo estendido, em branco, para a construção de um fragmento de ontologia.

Tabela 26 – Especificação parcial do padrão *Resource Nature* (RN) (SOUZA et al., 2016)

Characteristic	Description
Name	Resource Nature/RN
Intent	To represent resource types in terms of shareability.
Rationale	A resource can be either sharable or exclusive.
Competency Questions	- What is the nature (in terms of shareability) of a given resource? - Which resources are shareable? - Which resources are exclusive?
Axioms	- Exclusive Resource(r) $\rightarrow \forall p, p'$ participation_of(p, r) \wedge participation_of(p', r) $\wedge p \neq p' \rightarrow \neg$ (overlaps($t_i(p), t_i(p')$)), where t_i is a given time interval; - Sharable Resource(r) $\rightarrow \neg$ Exclusive Resource(r).

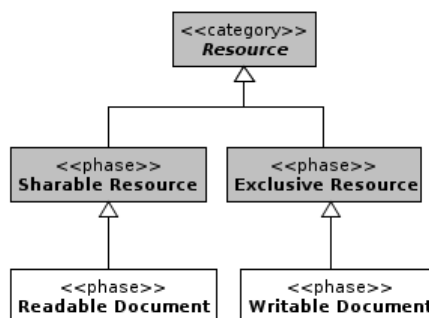


Figura 41 – Extensão do DROP RN adaptado (SOUZA et al., 2016)

4.4 Validar Padrões

Quando o conjunto de padrões estiver pronto ou ao menos bastante adiantado, os padrões criados devem ser validados. A validação de padrões, bem como a de ontologias como um todo, é ainda uma questão de pesquisa em aberto. Entretanto, esta abordagem adota a validação por exemplo como uma maneira de testar a viabilidade dos padrões na geração de fragmentos ontológicos adequados.

Propõe-se que cada padrão seja testado com vários exemplos para dar uma ideia se são flexíveis o bastante para se adequar a vários contextos de modelagem no domínio em questão. O capítulo 6 desta dissertação ilustra, por meio de estudos de casos, como a abordagem de padrões pode ser utilizada e, como consequência, valida alguns padrões a partir desses mesmos casos.

4.5 OPL de Colaboração

Uma vez extraídos e validados os DROPs, o conjunto formado por eles, juntamente com suas especificações, forma um catálogo de DROPs. Isso por si só, já é de grande auxílio para os engenheiros de ontologias, mas, ainda assim, o uso de catálogos exige algum esforço, para que se identifique que padrão utilizar em que dado momento da criação de uma nova ontologia. Por exemplo, Falbo et al. (2016) argumenta que organizar OPs em catálogos não é suficiente, uma vez que não abordam o forte senso de conexão entre os padrões. Este problema é particularmente saliente no caso dos chamados OPs de Domínio (DROPs), uma vez que esses padrões são muito inter-relacionados, sendo muito difícil aplicá-los isoladamente.

Para prover mais suporte ao utilizador dos padrões apresentados na seção 3.3.2 esta dissertação propõe uma OPL de Colaboração. Para prover mais suporte ao utilizador dos padrões apresentados na seção 3.3.2, esta dissertação propõe uma OPL de Colaboração. Como descrito na seção 2.5, uma OPL fornece um processo que descreve em que ordem utilizar os padrões para resolver problemas de modelagem do domínio a que se presta.

A OPL aqui proposta se chama C-OPL (*Collaboration Ontology Pattern Language*) e tem o objetivo de guiar os engenheiros de ontologias na utilização dos DROPs de Colaboração presentes no catálogo criado a partir da CONTO. O modelo do processo da C-OPL pode ser visto na Figura 42.

A Figura 42 apresenta C-OPL usando a linguagem de especificação de OPLs proposta por (QUIRINO, 2016) e foi construído com a utilização da ferramenta Astah¹ uma vez que o OLEDB ainda não suporta a criação de OPLs nele. De acordo com essa linguagem, i) os DROPs são representados por nós de ação (os retângulos arredondados

¹ <http://astah.net/>

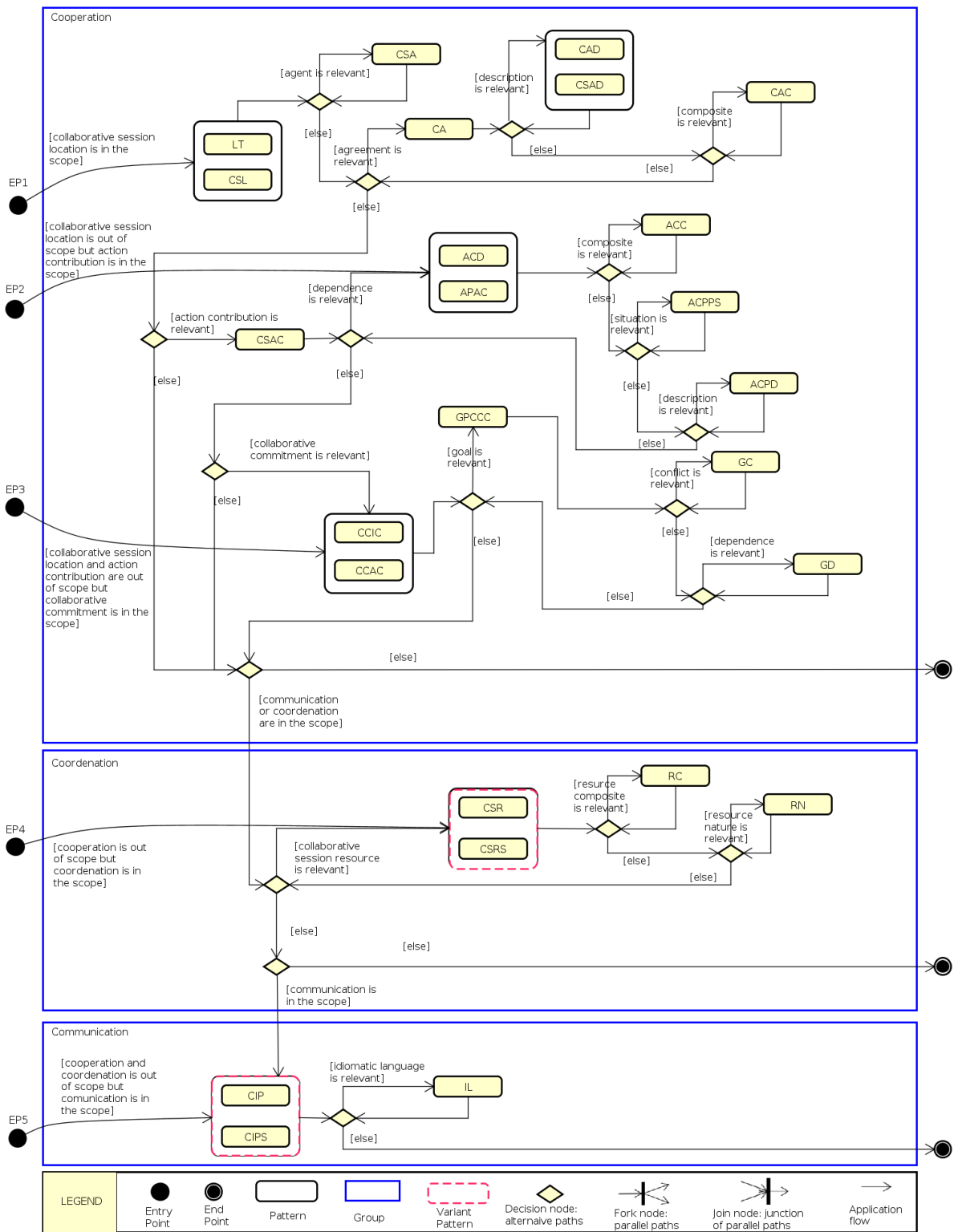


Figura 42 – Modelo do processo da C-OPL

rotulados); ii) os nós iniciais (círculos sólidos) representam pontos de entrada na OPL, ou seja, DROPs da linguagem que podem ser usados sem resolver outros problemas primeiro; iii) os nós finais (círculo sólido com auréola externa) representam pontos de saída na OPL, ou seja, DROPs da linguagem que podem ser usados para finalizar caminhos percorridos na linguagem; iv) os fluxos de controle (linhas com setas) representam as sequências admissíveis nas quais DROPs podem ser usados; v) os pontos de fusão/decisão (símbolos em forma de diamante) representam a mesclagem de caminhos na OPL (fusão) ou um ponto de tomada de decisão em um caminho (decisão); vi) os nós junção/divisão (segmentos de linha) representam a conjunção de caminhos (junção) ou caminhos independentes e paralelos (divisão); e vii) uma extensão da notação UML original (linhas pontilhadas com setas) é usada para representar padrões variantes/alternativos, isto é, padrões que podem ser usados para resolver o mesmo problema de diferentes maneiras.

O ponto de partida na utilização de uma OPL se dá pelos pontos de entrada que o modelo do processo da linguagem. A C-OPL tem cinco pontos de entrada. O engenheiro de ontologias deve escolher um deles, dependendo do escopo da ontologia de colaboração em desenvolvimento. EP1 deve ser escolhido quando os requisitos para a nova ontologia incluírem a definição de onde acontecerá a colaboração. Já EP2 deve ser escolhido quando o objetivo for especificar as ações de contribuição sem se importar onde ocorrerão (EP1 é quem se importa com isso). EP3 é a opção para quando o interesse estiver nos compromissos de colaboração sem se importar com as ações de contribuição (EP2) e onde elas acontecem (EP1). EP4 deve ser escolhido quando os requisitos incluírem a coordenação no processo de colaboração versando sobre tipos de recursos e modelando a participação desses recursos na seção colaborativa. Por último, EP5 quando os requisitos incluírem a definição da comunicação durante a colaboração sem interesse em como a cooperação se procederá e como a coordenação será feita. EP1, EP2 e EP3, são pontos de entrada quando o objetivo para a nova ontologia incluir cooperação; EP4, se incluírem a coordenação e EP4, quando incluírem comunicação sem cooperação.

Provavelmente, se perceberá, pelo modelo do processo da C-OPL (Figura 42), a ausência da definição de outros DROPs que lidem com problemas de cooperação, coordenação e comunicação. Isso se deve ao fato de que a C-OPL ser um trabalho em andamento e, portanto, não finalizada. Logo, a C-OPL ainda está incompleta em sua definição e em estado de pesquisa e alteração, o que deve gerar novas versões com aprimoramentos por meio de refatoração de DROPs e especificação de novos. Em um futuro próximo, os DROPs de outros DROPs se juntarão aos de Cooperação, Coordenação e Comunicação definidos, fechando uma versão completa da linguagem.

5 Suporte Automatizado para Catálogos de Padrões Ontológicos

A fim de apoiar o gerenciamento e a utilização de catálogo de DROPs, e como uma das contribuições desta pesquisa, o OLED foi estendido com um novo módulo que inclui funcionalidades para a edição e reutilização de OP, ou seja, para criação de catálogo de DROPs e para utilização destes catálogos posteriormente. Em seu desenvolvimento, optou-se por estender um editor de ontologias existente, para aproveitar suas ferramentas de modelagem, verificação e validação, apresentadas na seção 2.3. Assim, o engenheiro de ontologias fica com uma solução completa, partindo da modelagem de ontologias a partir da extensão das DROPs e podendo, com o uso do sistema, verificá-la sintática e semanticamente, além de validá-la (no OLED, a partir da verificação de instâncias da ontologia).

Após um extenso trabalho de análise do OLED, definiram-se os seguintes requisitos para a implementação do editor de catálogo de OPs:

- (i) Todo projeto no OLED deve ser um potencial projeto de catálogo de DROPs;
- (ii) Todos os diagramas presentes no projeto OLED podem conter DROPs;
- (iii) Permitir a criação e a utilização de catálogos, usando a interface de modelagem já existente no OLED;
- (iv) Permitir a importação dos catálogos criados para utilização;
- (v) Os OPs criados devem ser passíveis de verificação e validação pelos recursos presentes na ferramenta;
- (vi) Permitir a configuração do catálogo de DROPs, bem como dos padrões que pertencem a ele;

5.1 Criação de Catálogos de Padrões Ontológicos

O ponto-chave na criação de catálogo de DROPs está na definição dos padrões que compõem o catálogo. Logo, para isso, o domínio específico de construção do catálogo é parte preponderante porque, sem um domínio, não se tem um escopo de aplicação do catálogo. Com o módulo implementado no OLED, os OPs podem ser criados de duas formas no OLED: a partir de uma ontologia de referência ou a partir do zero. As seções 5.1.1 e 5.1.2, respectivamente, descrevem cada uma delas.

5.1.1 Criando um OP a partir de uma Ontologia de Referência

Quando se tem e se quer utilizar uma ontologia de referência de domínio específico para a construção de um catálogo de DROPs, ela pode ser aproveitada para extrair os OPs para formação do catálogo. Uma das funcionalidades do módulo é a que possibilita a extração de um OP a partir de uma ontologia de referência. Nela, os OPs extraídos da ontologia de referência dão origem aos diagramas no projeto que representarão e conterão as classes e as relações pertencentes ao OP.

A Figura 43, pode-se ver como criar um OP a partir de uma ontologia de referência. Nela, é apresentado um fragmento da ontologia de cooperação a partir do qual é criado um OP (*RN*) com as classes *Resource*, *Sharable Resource* e *Exclusive Resource* juntamente com as respectivas relações de generalização. Na Figura 43, foram destacados os seguintes elementos:

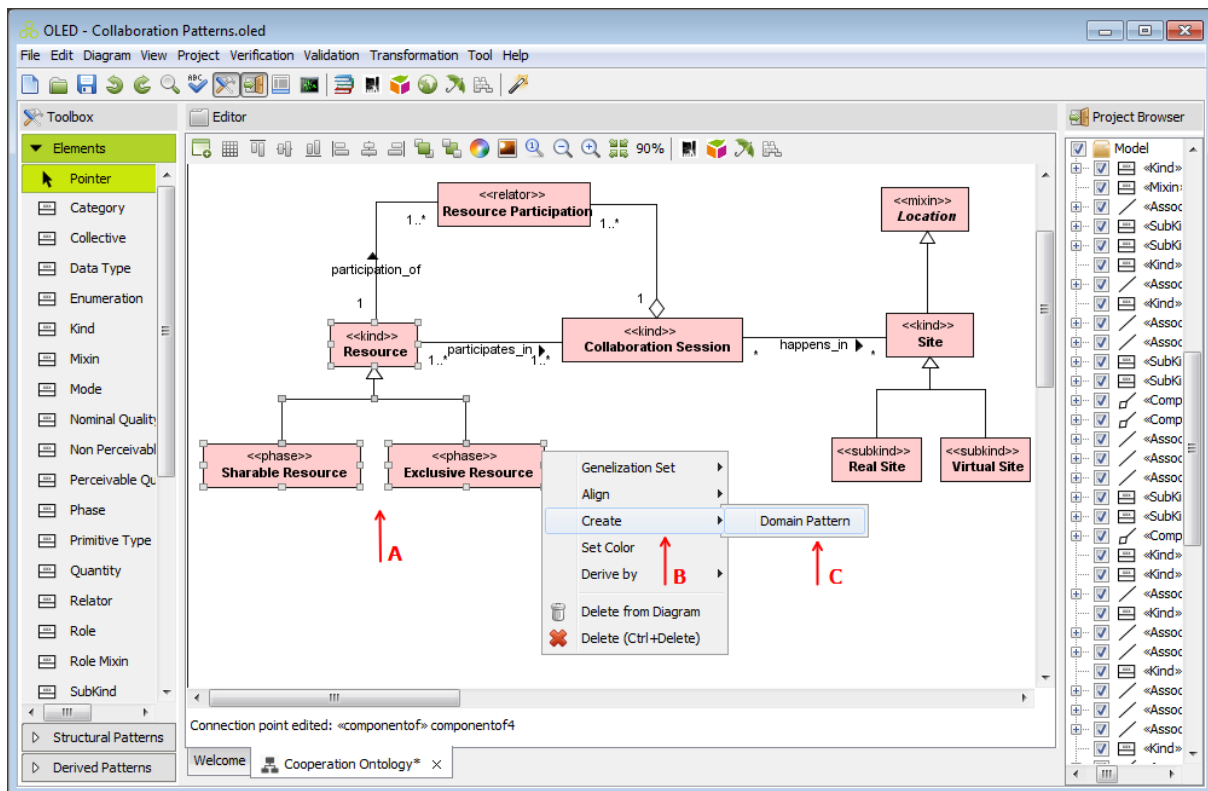


Figura 43 – Criação de um OP (*RN*) a partir de uma ontologia de referência

- (A): Classes OntoUML que formarão o OP deseja. Neste caso, o OP *RN* está sendo extraído do fragmento da ontologia referência de cooperação;
- (B) e (C): Opção no menu suspenso *Create* -> *Domain Pattern* acessada a partir do clique direito do *mouse* sobre um dos elementos selecionados no diagrama para a extração e criação de um OP (*RN*) a partir desses elementos;

Logo após a opção *Create -> Domain Pattern* ser clicada, será mostrada a janela para a inserção do nome do novo diagrama. Diagrama esse que conterá o fragmento selecionado da ontologia, ou seja, o OP. Na Figura 45 é visto a janela, que é a mesma apresentada na seção anterior, para a inserção do nome do diagrama. Uma vez informado o nome do diagrama (*RN*) na tela apresentada na Figura 45, o OP é criado conforme Figura 46. Basicamente, a diferença entre se criar um OP a partir de uma ontologia de referência ou do zero, está no ponto inicial de construção do OP. Quando a ontologia de referência existe, o processo fica mais rápido, menos moroso, uma vez que as classes já estão definidas com as respectivas relações, bastando apenas a seleção de quais formam os OPs desejados.

Quando não existe a ontologia de referência ou se deseja criar os OP do zero, todas as classes que farão parte do OP devem ser selecionadas manualmente da caixa de ferramentas do OLED (*toolbox*) ou aproveitá-las, caso já tenham sido utilizadas em outro OP do projeto, selecionando-as, também manualmente, no nó *Model* da árvore de navegação *Project Browser* e adicionando-as ao diagrama. A partir desse ponto, ambas as formas levam ao mesmo caminho, ou seja, a configuração do catálogo é a mesma independente da forma de construção de catálogo de DROPs. Quando a ontologia de referência é conhecida, o trabalho do engenheiro de OPs é facilitado com essa opção pois ele pode selecionar na ontologia de referência as classes que formam um OP desejado e criar o OP a partir deles.

5.1.2 Criando um OP a partir do Zero

Quando não há uma ontologia de referência do domínio para construir um catálogo de DROPs, os OPs deverão ser obtidos a partir das classes OntoUML implementadas no OLED. Dessa forma, o processo de construção de um catálogo de DROPs fica mais árduo, moroso e mais dependente do conhecimento de um especialista no domínio dos padrões. Isso significa que cada OP será desenvolvido criando-se o diagrama que o representa e adicionando-se, manualmente, cada elemento OntoUML (classe) que o compõe. A Figura 44 destaca os elementos participantes da criação do OP a partir do zero. São eles:

- (A): Menu de acesso à opção de criação de um novo diagrama no projeto. Esse novo diagrama será inserido como um nó folha do nó *Diagrams* (mostrado em *E*);
- (B) e (C): Classes OntoUML para a criação de ontologias e, neste caso, criação de OPs. Como exemplo, a Figura 46 mostra as classes que formam o OP *RN*;
- (D): Diagrama do projeto que conterá o OP. Para desenhar as classes deve-se selecioná-las com o mouse na caixa de ferramentas;
- (E): Nó da árvore de navegação do projeto que contém os diagramas do projeto. Nele

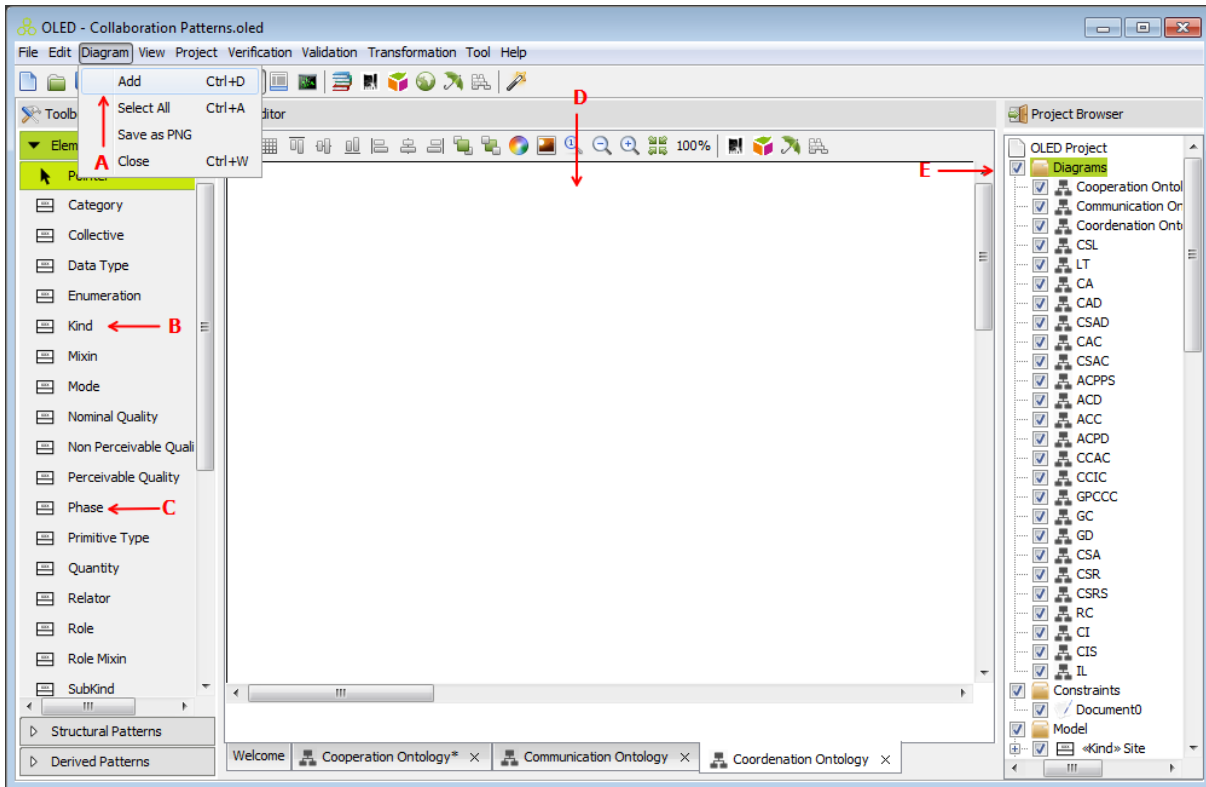


Figura 44 – Criação de um OP a partir do zero

também é possível clicar com o botão direito do *mouse* para acessar um menu suspenso com a opção de criação de um novo diagrama no projeto.

Para exemplificar, considere a criação do OP *RN*. Para criá-lo, deve-se criar um diagrama com o nome *RN*, conforme mostra a Figura 45. Ele será adicionado como filho do nó *Diagrams*. A Figura 45 destaca, ainda, alguns elementos que têm as seguintes funções:

- (A): Campo de entrada de dados para a informação do nome do diagrama (neste caso, *RN*);
- (B): Botão *OK* para a confirmação do nome e prosseguimento com a criação do novo diagrama;
- (C): Botão *Cancel* para o cancelamento da ação de criação de um novo diagrama;

Na Figura 46, encontra-se o OP *RN* e, em destaque, tem-se:

- (A): OP *RN* com as classes OntoUML que o compõem, *Resource*, *Sharable Resource* e *Exclusive Resource* juntamente com as respectivas relações de generalização;
- (B): Nó *Diagrams* da árvore de navegação, mostrando os diagramas do projeto que, em destaque, está o recém-criado nó *RN* (mostrado em *C*);

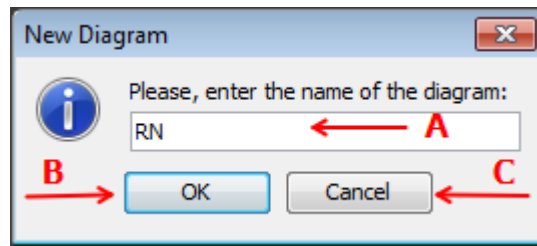


Figura 45 – Criação de um diagrama (RN)

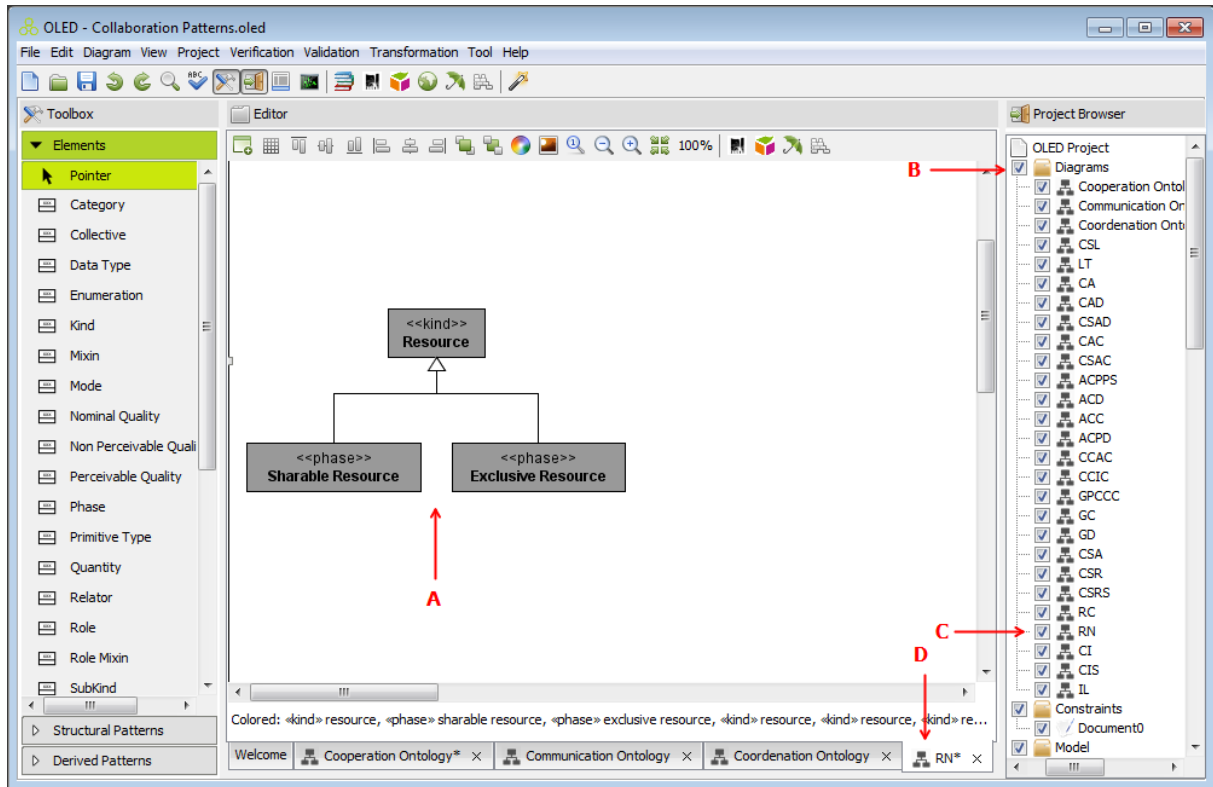


Figura 46 – OP RN

- (C) e (D): Visualização do diagrama RN com o OP criado;

5.1.3 Configuração de um Catálogo de OPs

Uma vez criados os diagramas com os respectivos OPs, as configurações do projeto podem ser acessadas para transformá-lo em um catálogo do DROPs. Isso pode ser feito clicando com o botão direito do *mouse* sobre o nó raiz da árvore de navegação do projeto conforme Figura 47. Nela, estão destacados os seguintes elementos:

- (A): Nó raiz da árvore de navegação do projeto (*OLED Project*) que, quando clicando com o botão direito do *mouse* mostrará um menu suspenso com a opção de configuração do catálogo de OPs;
- (B): Opção *Settings* para acesso as configurações do projeto como catálogo de OPs;

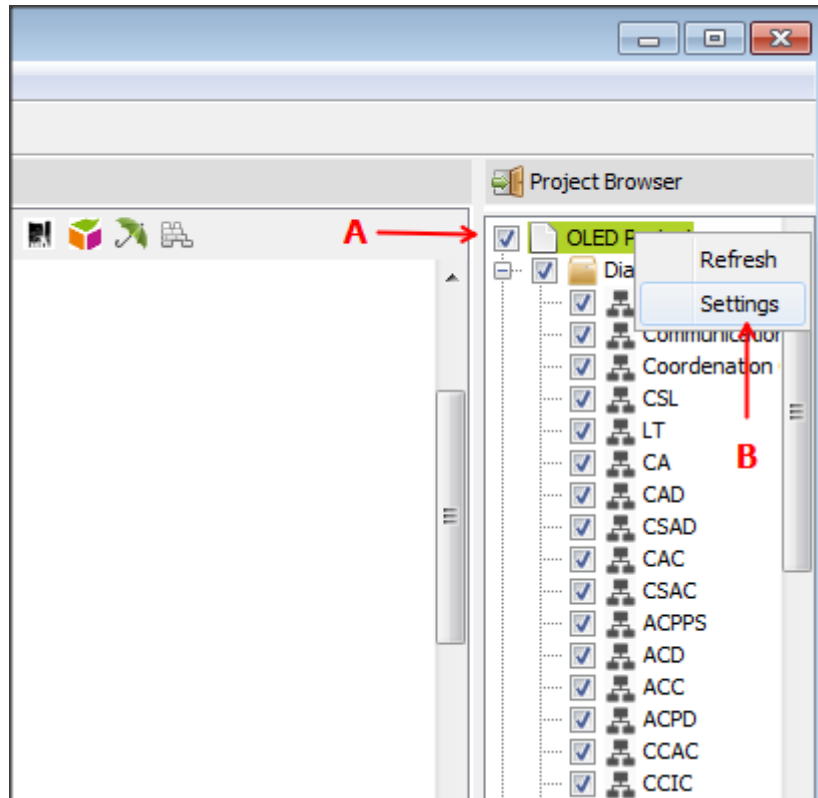


Figura 47 – Menu de acesso as configurações do catálogo de OPs

Com o acesso as configurações para o catálogo de DROPs do projeto, tem-se o acesso a tela apresentada na Figura 48 com os elementos participantes da configuração do projeto como catálogo de DROPs. Nela estão destacados alguns elementos que tem as seguintes funções:

- (A): Botão *checkbox Is pattern project?* para configurar o projeto como catálogo de OPs;
- (B): Campo *Pattern catalogue name* de entrada para informar o nome do catálogo;
- (C): *Label* contendo o nome do arquivo que contém a representação visual do encadeamento dos DROPs do catálogo, ou seja, o processo dos OPs do catálogo;
- (D): Botão *Preview* para visualizar o conteúdo do arquivo selecionado no item C. Um exemplo pode ser visto na Figura 49 em foi definido o processo para o catálogo de DROPs de serviço;
- (E): Botão *Choose* que abrirá uma tela de seleção para o arquivo que contém a representação visual do encadeamento dos DROPs do catálogo. Uma vez tendo sido selecionado um arquivo, o *label* do botão passará para *Clear* que, quando clicado, limpará os dados do arquivo e o *label* voltará para *Choose*, assumindo a função de escolha inicial;
- (F): Tabela *Diagram Pattern Mapping* contendo a listagem dos diagramas e o mapeamento para os OPs;

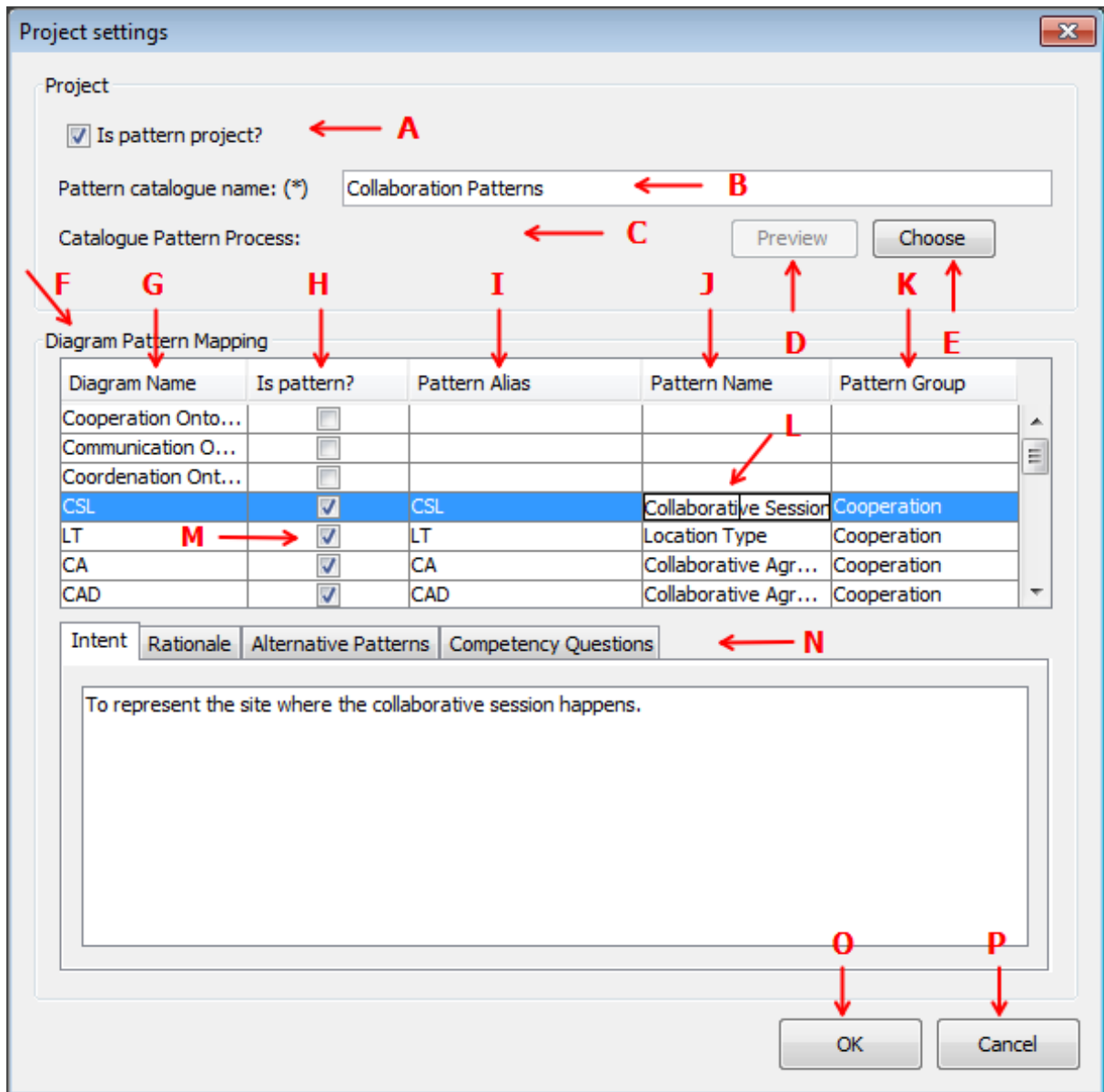


Figura 48 – Configuração do projeto OLED como catálogo de OPs

- (G): Coluna *Diagram Name* em que é mostrado o nome dos diagramas pertencentes ao projeto;
- (H): Coluna *Is pattern?* em que é mostrado um botão *checkbox* de configuração do diagrama como um OP. Caso ele seja marcado, o restante das colunas daquele registro ficará disponível para edição;
- (I): Coluna *Pattern Alias* para informar um apelido para o padrão, ou seja, o nome pelo qual ele será conhecido no catálogo. Edição disponível por meio do duplo clique na célula correspondente para que a mesma se torne editável;
- (J): Coluna *Pattern Name* para informar um nome para o padrão. Da mesma forma, basta um duplo clique na célula correspondente para que a mesma se torne editável;

- (K): Coluna extitPattern Group para informar um nome de grupo para o OP. Outra vez, basta um duplo clique na célula correspondente para que a mesma se torne editável;
- (L): Célula para a informação do nome do padrão em modo edição;
- (M): Botão *checkbox* selecionado indicando que o respectivo diagrama é um OP e que as informações do OP estão disponíveis para edição;
- (N): Abas para o preenchimento das características do OP selecionado;
- (O): Botão *OK* para confirmar as alterações de configuração do catálogo;
- (P): Botão *Cancel* para cancelar as alterações de configuração do catálogo;

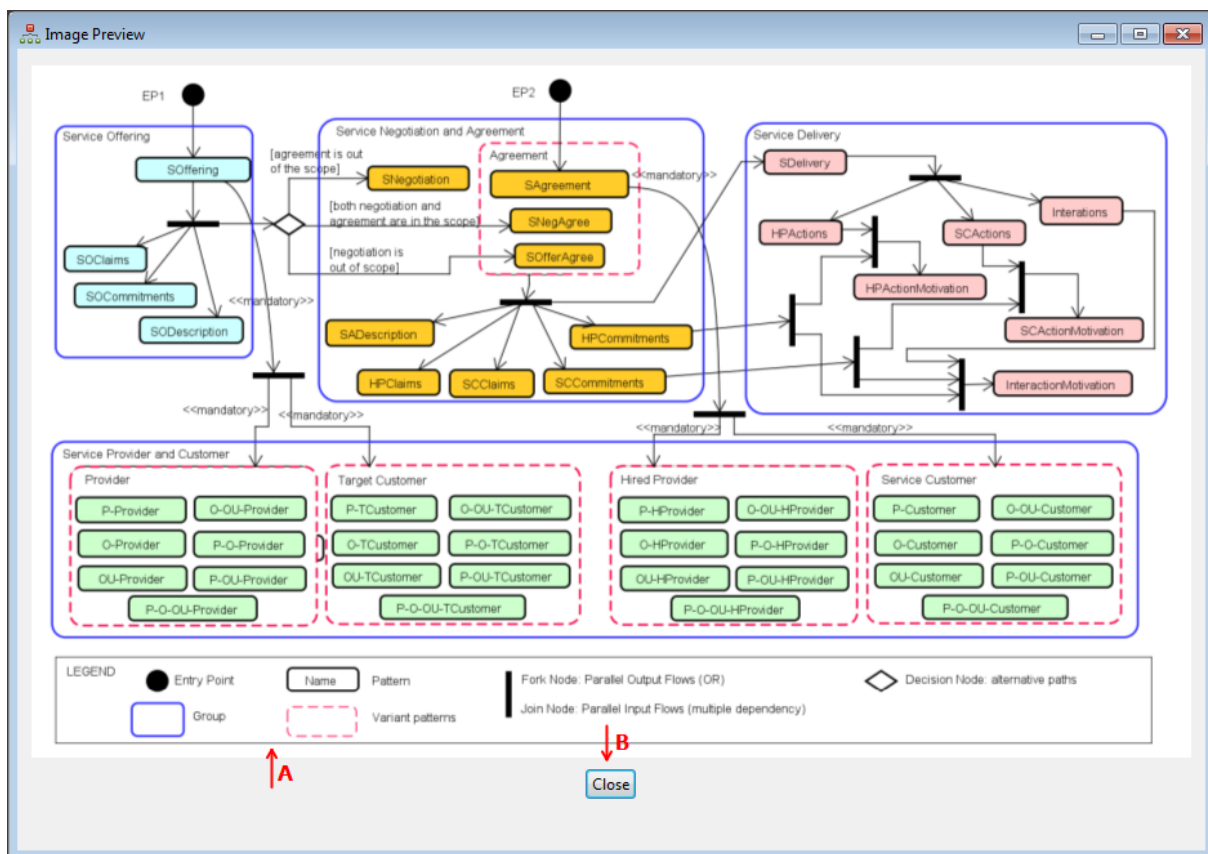


Figura 49 – Visualização do conteúdo do modelo do processo

Na tela de visualização do processo modelado para os DROPs pertencentes ao catálogo, tem-se, em destaque, os seguintes elementos:

- (A): Painel que que é mostrado o processo de seleção dos DROPs do catálogo. Neste caso, a definição para o catálogo de DROPs de serviço pode ser visto;
- (B): Botão *Close* para fechar a tela;

5.1.3.1 Intenção

A Intenção está ligada ao propósito do OP, ou seja, o que o projetista do OP quer que o OP cumpra dentro do domínio. Isso pode ser resumido pela pergunta *O quê?*. A intenção é o ponto de partida, é o fator motivador para a definição de um OP porque é ela que nos diz o que se quer fazer. A organização das intenções forma a base para desencadeamento de um projeto de uma ontologia de referência que por sua vez, conterá os OP que formarão o catálogo de DROPs.

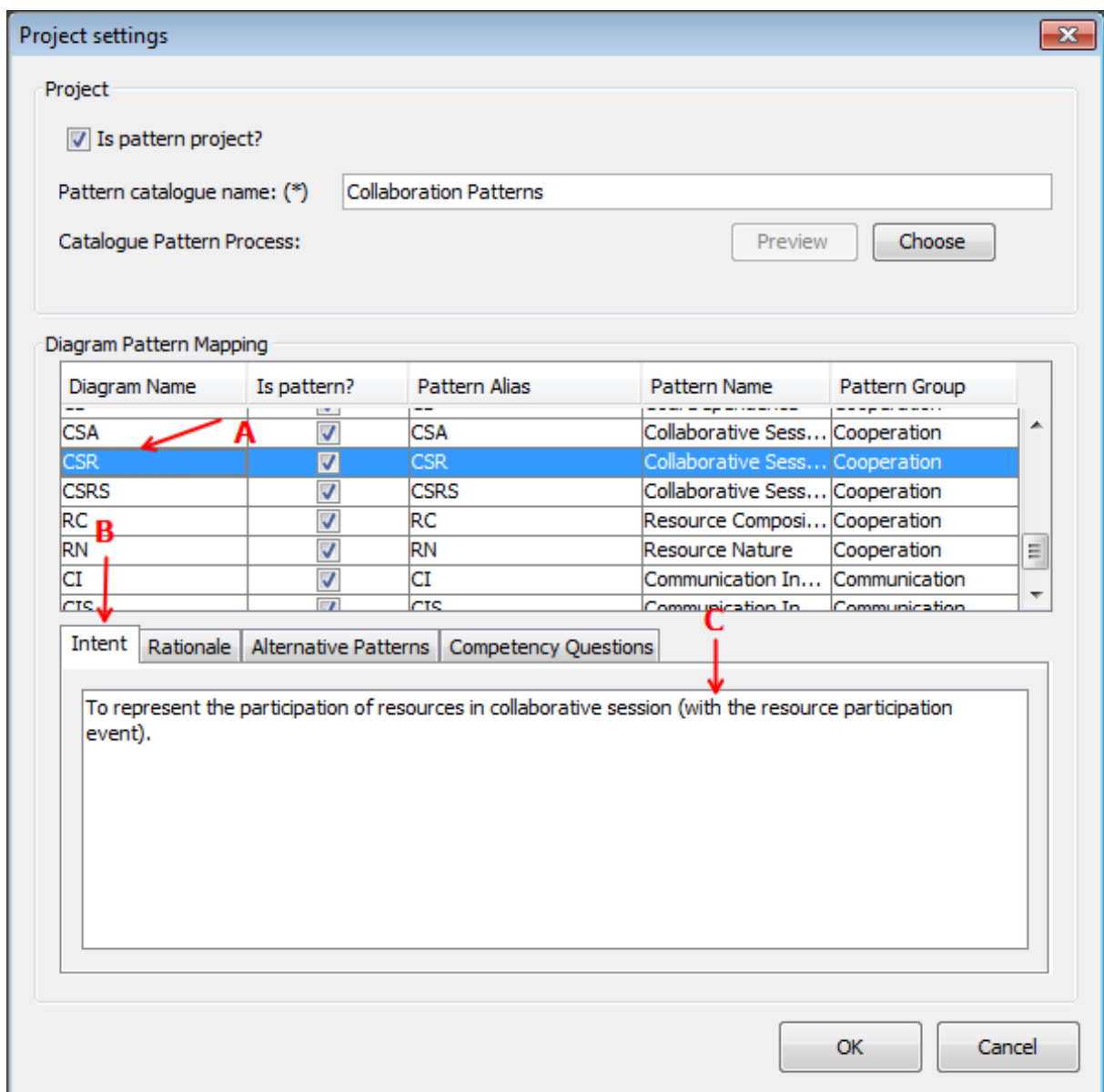


Figura 50 – Configuração da intenção do catálogo de OPs

Na Figura 50, é apresentada a aba *Intent* selecionada com o elemento participante da configuração da intenção do OP selecionado. Nela, pode-se ver destacados alguns elementos que tem as seguintes funções:

- (A): Registro da tabela de mapeamento entre diagrama e padrões em que o OP *CSR*, neste caso, está selecionado como pode ser visto pelo destaque do registro;
- (B): Aba *Intent* selecionada apresentando o elemento de configuração para a intenção;
- (C): Campo responsável por permitir a entrada da descrição da intenção do OP;

5.1.3.2 Justificativa

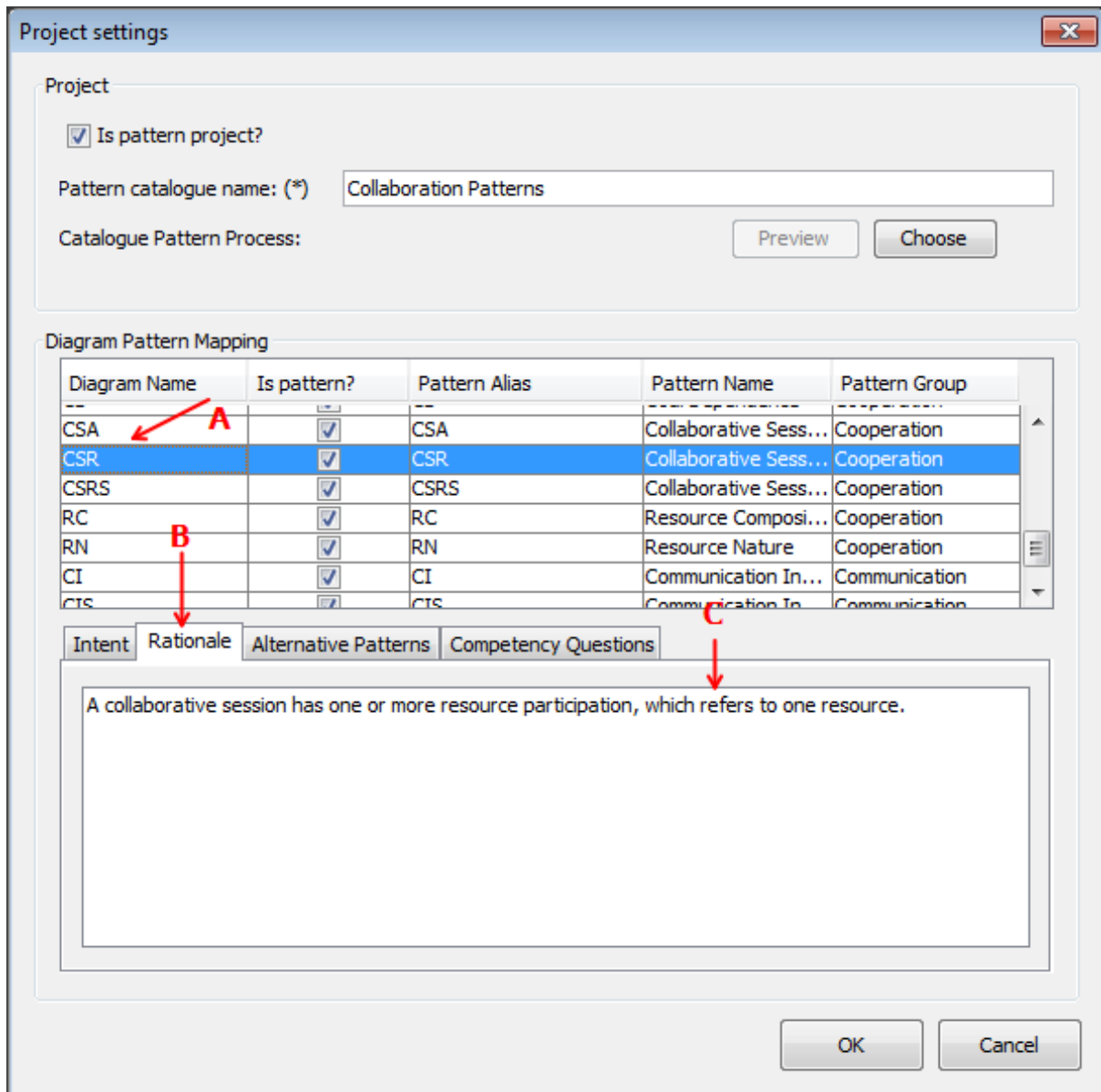


Figura 51 – Configuração da justificativa do catálogo de OPs

A justificativa auxilia o modelador para possa compreender o raciocínio lógico por trás da definição do OP. Como, muitas vezes, o modelador não é um especialista no domínio em que o catálogo de OPs foi desenvolvido, ele pode entender, por meio desta informação, o porquê do padrão ter sido definido. Isso também permite a aquisição de algum conhecimento do domínio específico.

Na Figura 51, é vista a aba *Rationale* selecionada com os elementos participantes da configuração da justificativa do OP selecionado. Nela estão destacados alguns elementos que tem as seguintes funções:

- (A): Registro da tabela de mapeamento entre diagrama e padrões em que o OP *CSR*, neste caso, está selecionado como pode ser visto pelo destaque do registro;
- (B): Aba *Rationale* selecionada apresentando o elemento de configuração da justificativa;
- (C): Campo responsável por permitir a entrada da descrição da justificativa do OP;

5.1.3.3 Padrões Alternativos

As relações entre os padrões do catálogo de OPs são muito importantes para que o modelador possa saber o tipo de relacionamento existente entre eles ajudando, assim, na escolha dos mesmos para construção de uma ontologia. Muitas vezes, ao se modelar uma ontologia de referência, tem-se partes dela que podem resolver problemas de forma diferente dependendo do que a solução necessita. Por exemplo, o tipo de relação em que um OP é alternativo a outro, uma vez que ambos lidam com o mesmo problema, mas de forma diferente, por exemplo, a complexidade.

Na Figura 52, pode-se ver a aba *Alternative Patterns* selecionada com os elementos participantes da configuração dos relacionamentos do OP selecionado. Nela estão destacados alguns elementos que tem as seguintes funções:

- (A): Registro da tabela de mapeamento entre diagrama e padrões em que o OP *CSR*, neste caso, está selecionado como pode ser visto pelo destaque do registro;
- (B): Aba *Alternative Patterns* selecionada apresentando os elementos de configuração dos relacionamentos, no caso, relacionamento alternativo em que um OP pode ser substituído por outro, conforme mencionado anteriormente;
- (C): Botões *checkbox* que são responsáveis pela configuração de OPs alternativos;
- (D): Coluna *Pattern name* que é responsável por apresentar os OPs presentes no catálogo em que cada OP é um registro na tabela de configuração do relacionamento;
- (E): Coluna *Is alternative* que é responsável por disponibilizar um botão *checkbox* para a configuração do OP do registro como alternativa do OP selecionado, ou seja, podem ser trocados um pelo outro dependendo das características que o modelador necessitar;
- (F): Botões *checkbox* selecionado mostrando que o OP *CSRS*, neste caso, é uma OP alternativo;

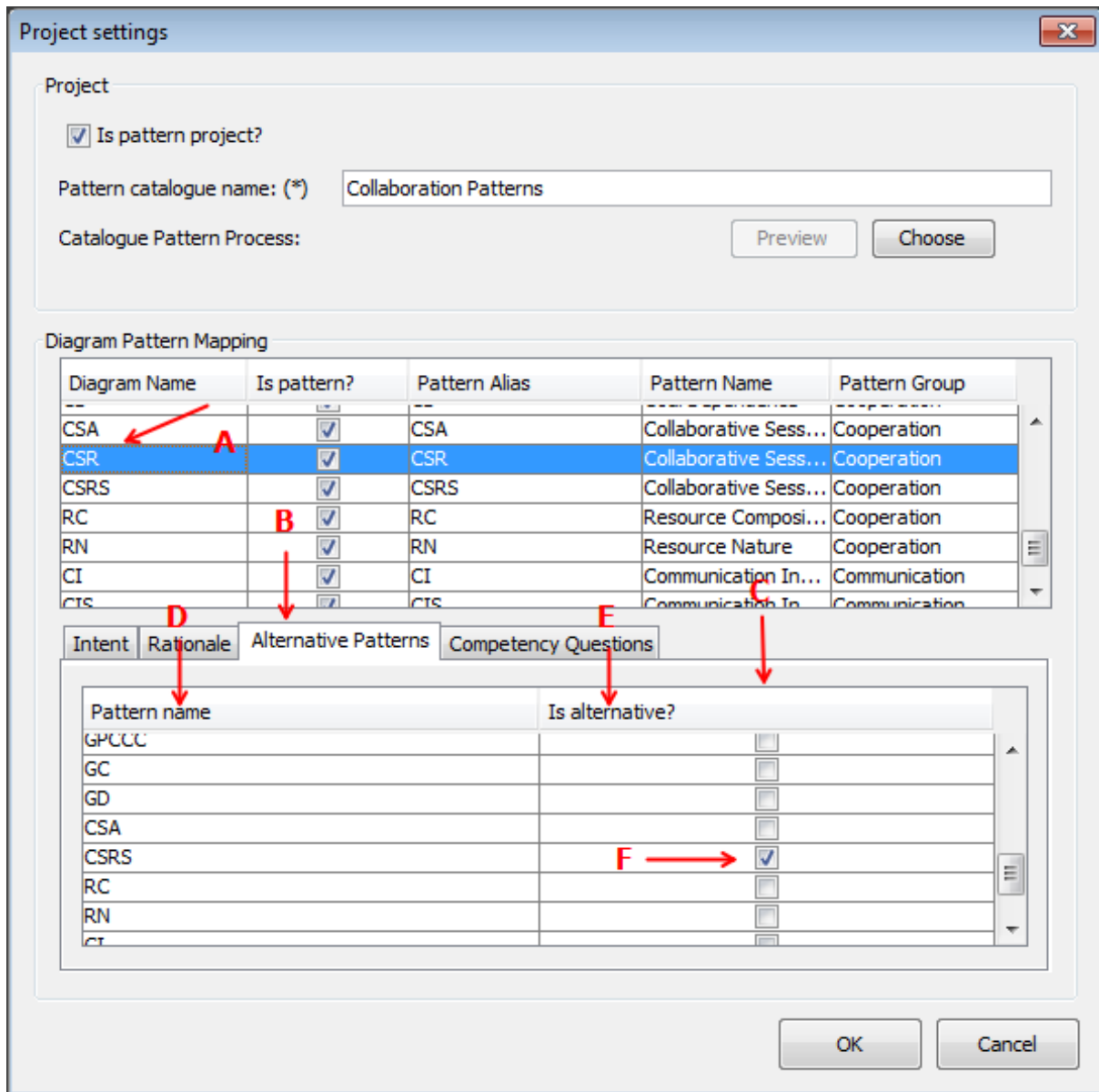


Figura 52 – Configuração do relacionamento entre os OPs do catálogo de OPs

5.1.3.4 Questões de Competência

As CQs são de fundamental importância pois são elas que nos respondem à pergunta "qual é o problema recorrente que o OP resolve?". As definições delas aqui que foram feitas utilizando a metodologia SABIO (FALBO, 2011). Os OPs têm por objetivo responder/lidar com algum problema recorrente de domínio específico mapeado (BUSCHMANN; HENNEY; SCHMIDT, 2007). Então, na definição de um catálogo, tem-se a opção de serem informados quais foram as questões que levaram a construção dos OPs e, automaticamente, qual(is) problema(s) recorrente(s) que ele procura resolver. As questões de competência informadas aqui poderão ser visualizadas pelo modelador quando ele posicionar o ponteiro do *mouse* sobre o OP do catálogo quando este tiver sido importado no OLED.

Na Figura 53, é mostrada a aba *Competency Questions* selecionada com os elementos

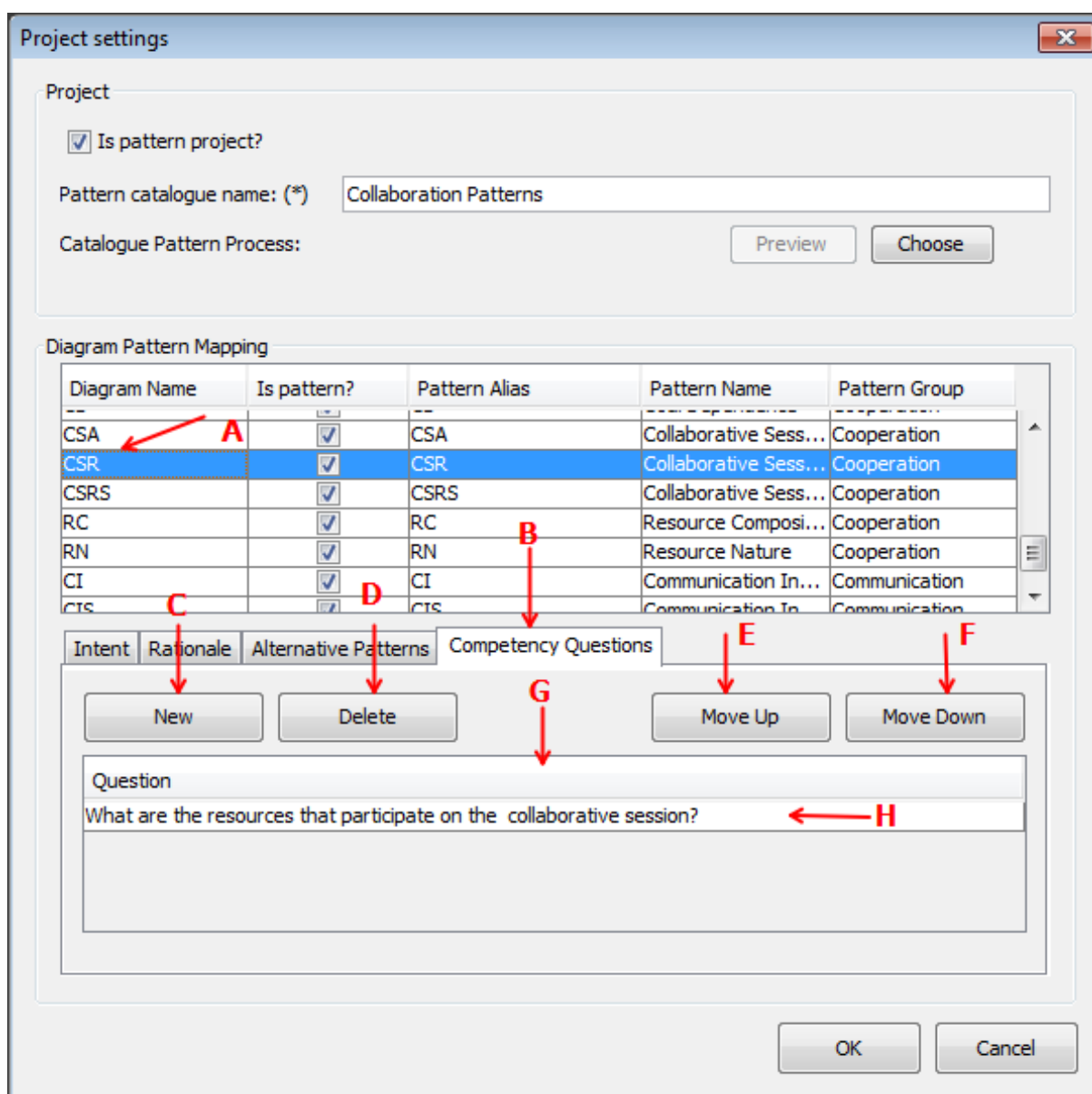


Figura 53 – Configuração das CQs do OP

participantes da configuração das CQs do OP selecionado em que estão destacados alguns elementos que tem as seguintes funções:

- (A): Registro da tabela de mapeamento entre diagrama e padrões em que o OP *CSR*, neste caso, está selecionado como pode ser visto pelo destaque do registro;
- (B): Aba *Competency Questions* selecionada apresentando os elementos de configuração das CQs;
- (C): Botão *New* que é responsável por incluir um novo registro em branco na listagem de CQs (G) que se torna editável com um duplo clique do *mouse*, permitindo assim a inclusão do conteúdo;

- (D): Botão *Delete* é responsável por excluir um registro da listagem de CQs (*G*) bastando selecioná-lo na listagem de CQs (*G*) antes de clicar sobre o botão *Delete* (*D*) para que a CQ seja excluída da listagem;
- (E): Botão *Move Up* que é responsável por mover um registro da listagem de CQs (*G*) para cima bastando, para isso, selecionar a CQ na listagem antes do clique;
- (F): Botão *Move Dow* que é responsável por mover um registro da listagem de CQs (*G*) para baixo bastando, para isso, selecionar a CQ na listagem antes do clique;
- (G): Lista *Question* com as CQs cadastradas para o padrão selecionado;
- (H): Registro na lista de CQs (*G*) de uma CQ. O Registro se torna editável com um duplo clique do *mouse*;

5.2 Utilização de Catálogos de Padrões Ontológicos

A utilização de catálogo de DROPs é uma forma de auxiliar e agilizar a criação de ontologias operacionais de domínio específico porque já disponibilizam ferramentas para solução de problemas recorrentes estudados e mapeados nesse domínio. Uma analogia para um catálogo de DROPs pode ser uma caixa de ferramentas composta por várias ferramentas pois nela estão contidas as ferramentas com maiores chances de utilização.

Como exemplo, podem ser citadas as chaves de boca de vários tamanhos, chaves de fenda de vários tamanhos, parafusadeiras de vários tamanhos, etc.... Cada uma dessas ferramentas foi criada para resolver um problema comum e, quando uma dessas chaves é necessária, não é necessário a criação da ferramenta uma vez que alguém, percebendo que elas resolvem problemas comuns, já as inventou e agrupou em uma caixa de ferramentas. Então, basta selecionar a chave ideal para o trabalho. Com o catálogo de OPs é a mesma coisa, alguém já identificou os OPs do domínio (as chaves) e agrupou em um catálogo (caixa de ferramentas) bastando, assim, selecionar os OPs necessários para construir a ontologia operacional.

Com esse objetivo, o OLED foi dotado de uma funcionalidade para a importação de catálogo de DROPs. Nas seções 5.2.1 e 5.2.2 são detalhados os passos para o carregamento e utilização de catálogo de DROPs no OLED.

5.2.1 Carregando um Catálogo de DROPs no OLED

Para utilização de catálogos de DROPs, eles devem ser importados no OLED. Nesse contexto, carregar e importar tem o mesmo sentido que é o de disponibilizar os DROPs de um catálogo para utilização no OLED. Para isso, foi desenvolvida uma funcionalidade no OLED que nos possibilita essa tarefa. Na Figura 55, pode ser visto os elementos

participantes do acesso a opção de importação do catálogo de OPs:

- (A), (B) e (C): *Menu* de acesso a opção de importação do catálogo de DROPs que é acessado em *File -> Import from -> Domain Pattern (OLED)*;

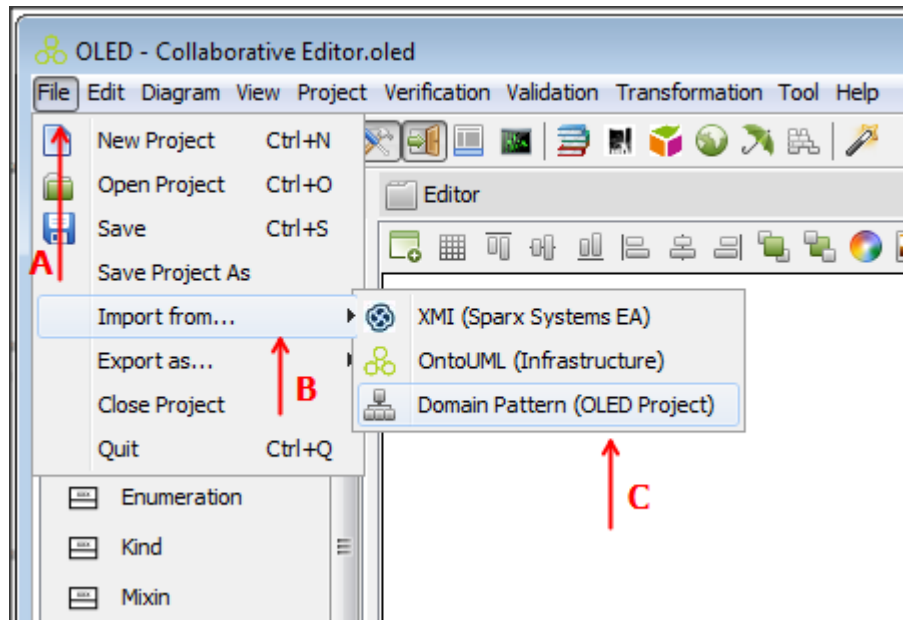


Figura 54 – Importação de um catálogo de OPs no OLED

Como foi definido um catálogo de DROPs para o domínio de colaboração no capítulo 3 (seção 3.3.2) e ele foi implementado no capítulo 5.1, então ele será utilizado para demonstrar como importar um projeto de catálogo de DROPs e utilizar os OPs presentes nele. Prosseguindo então, uma vez selecionado o item do *menu* para a importação do catálogo (Figura 54), será aberta uma janela para a seleção do projeto do catálogo, conforme Figura 55. Nela, são apresentados os elementos que compõem a tela com alguns destacados que executam as seguintes funções:

- (A): Arquivo com o projeto do catálogo de DROPs que se deseja importar no OLED;
- (B): Opção para a seleção do local em que se encontra o projeto com o catálogo (A);
- (C): Campo de entrada para a informação do nome do arquivo do projeto do catálogo que é preenchido automaticamente com a seleção do arquivo (A);
- (D): Botão *Open* de disparo da ação de importação do catálogo selecionado (A);
- (E): Botão *Cancel* de cancelamento da importação do arquivo o projeto do catálogo;

Caso a importação do catálogo de DROPs seja cancelada, a tela de seleção do catálogo de DROPs será fechada. Caso a importação seja confirmada após a seleção do

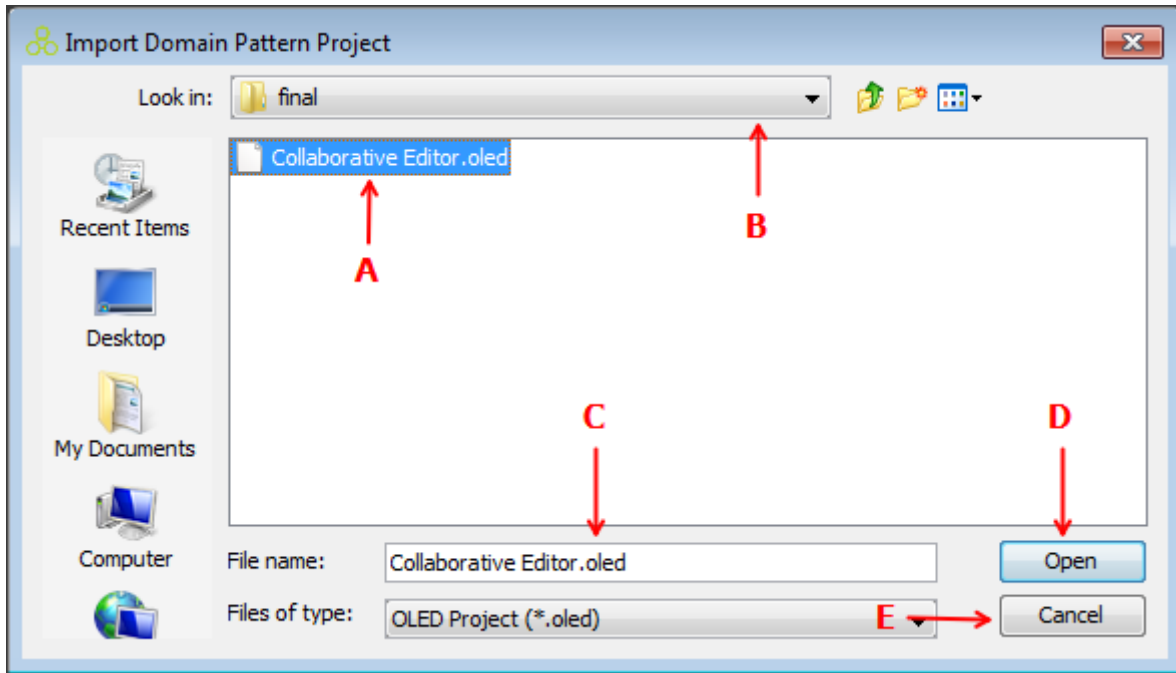


Figura 55 – Escolha de um catálogo de OPs no OLED

arquivo do catálogo, prosseguindo com a importação pelo acionamento do botão *Open* mostrado na Figura 55, os OPs serão carregados no OLED na caixa de ferramentas (*toolbox*), painel vertical esquerdo, e arranjados conforme apresentado na Figura 56 em que os elementos participantes do catálogo importado estão destacados e têm as seguintes funções:

- (A): Entrada na caixa de ferramentas (*toolbox*) do catálogo de DROPs, apresentando o nome do catálogo que foi definido na tela apresentada na Figura 48 (B);
- (B) e (C): Grupos de DROPs definidos e informados na definição do catálogo conforme na tela presente na Figura 48 (H);
- (D): OP (ACPD) pertencente ao catálogo de DROPs exemplificando os OPs do catálogo;
- (E): Balão, mostrado toda vez que o ponteiro do *mouse* for colocado sobre o OP, contendo informações resumidas do OP que são configuradas nas telas apresentadas nas Figuras 48, 50, 51, 52 e 53;

5.2.2 Utilizando um Catálogo de DROPs no OLED

Com o catálogo de DROPs carregado no OLED (seção 5.2.1), o modelador tem acesso aos OPs do catálogo para construção de ontologias operacionais de domínio específico. Uma das grandes vantagens da utilização de catálogo de DROPs para a modelagem de ontologia operacionais vem do fato de se ter os problemas, mais comumente encontrados no

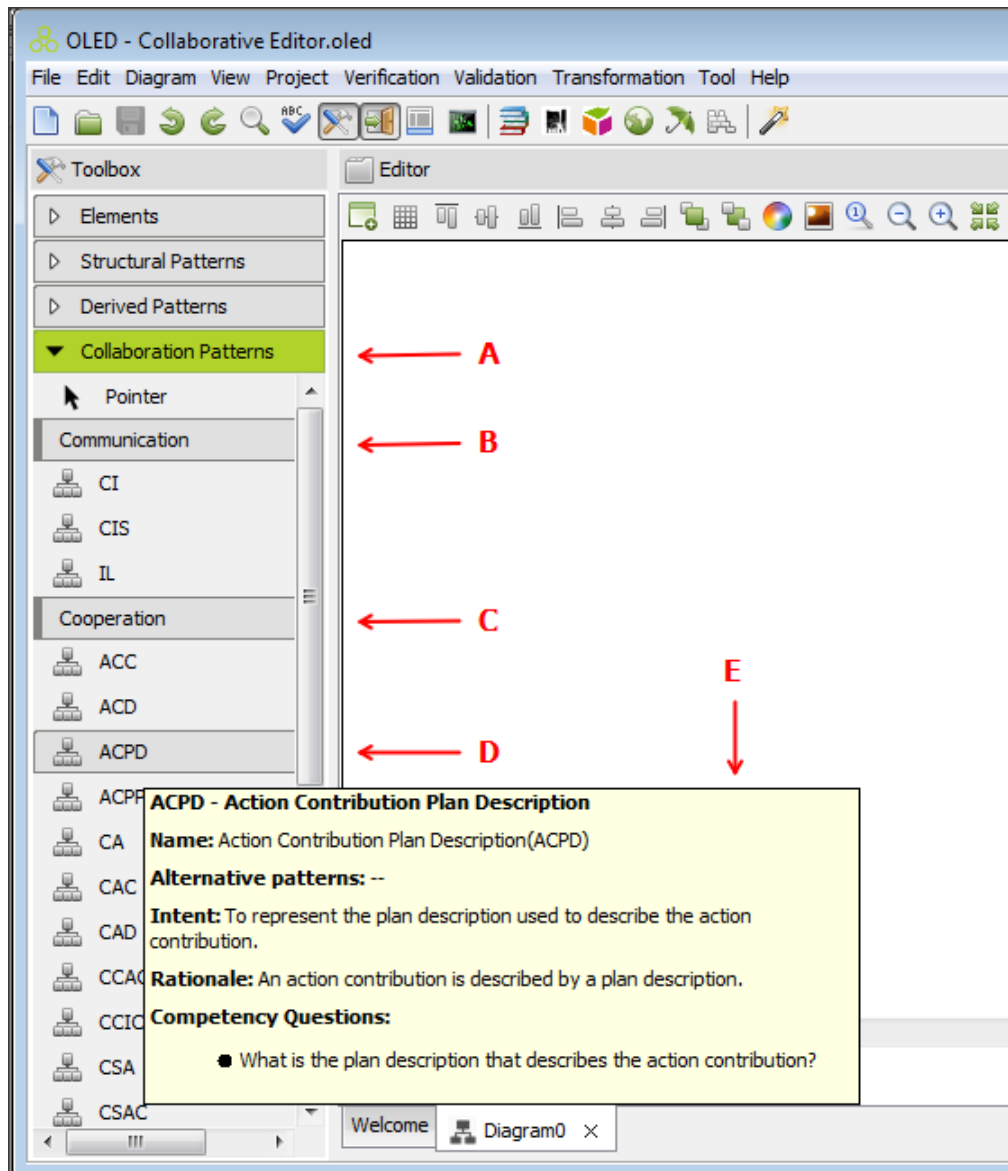


Figura 56 – Visualização de um catálogo de OPs carregado no OLED

domínio para o qual o catálogo foi concebido, mapeados e resolvidos pelos OPs do catálogo. Outra vantagem é que não é necessário que um especialista no domínio do catálogo esteja sempre presente para a elaboração das ontologias operacionais, desonerando, assim, a construção das mesmas.

Para demonstrar a utilização de OPs de catálogos, será utilizado o catálogo de DROPs no domínio de colaboração criado na no capítulo 5.1 e importado na seção 5.2.1 desse capítulo, ou seja, um catálogo de DROPs de colaboração.

Na Figura 57, tem-se o catálogo de DROPs colaborativos carregado no caixa de ferramentas (*toolbox*) do OLED em que foi selecionado o OP *CSR*. Também podem ser vistos outros elementos na tela presente da figura pertencentes a essa etapa do processo em destaque:

- (A): Entrada na caixa de ferramentas (*toolbox*) do catálogo de DROPs, apresentando o nome do catálogo que foi definido na tela apresentada na Figura 48 (B);
- (B): OP (*CSR*), pertencente ao catálogo de DROPs selecionado, para utilização na construção de uma ontologia operacional no domínio colaborativo;
- (C): Balão mostrando as informações resumidas do OP *CSR* que foram configuradas nas telas apresentadas nas Figuras 48, 50, 51, 52 e 53;
- (D): Área de trabalho representando o conteúdo do diagrama corrente em que é desenhada a ontologia, ou seja, área em que se deve clicar com o *mouse*, após a seleção do OP no catálogo (B), para a criação do OP;
- (E): Nó *Model* da árvore de navegação do projeto em que são adicionadas as classes OntoUML pertencentes ao projeto. As classes OntoUML pertencentes aos OPs do catálogo serão adicionadas nesse nó quando esses OPs forem utilizados no projeto;

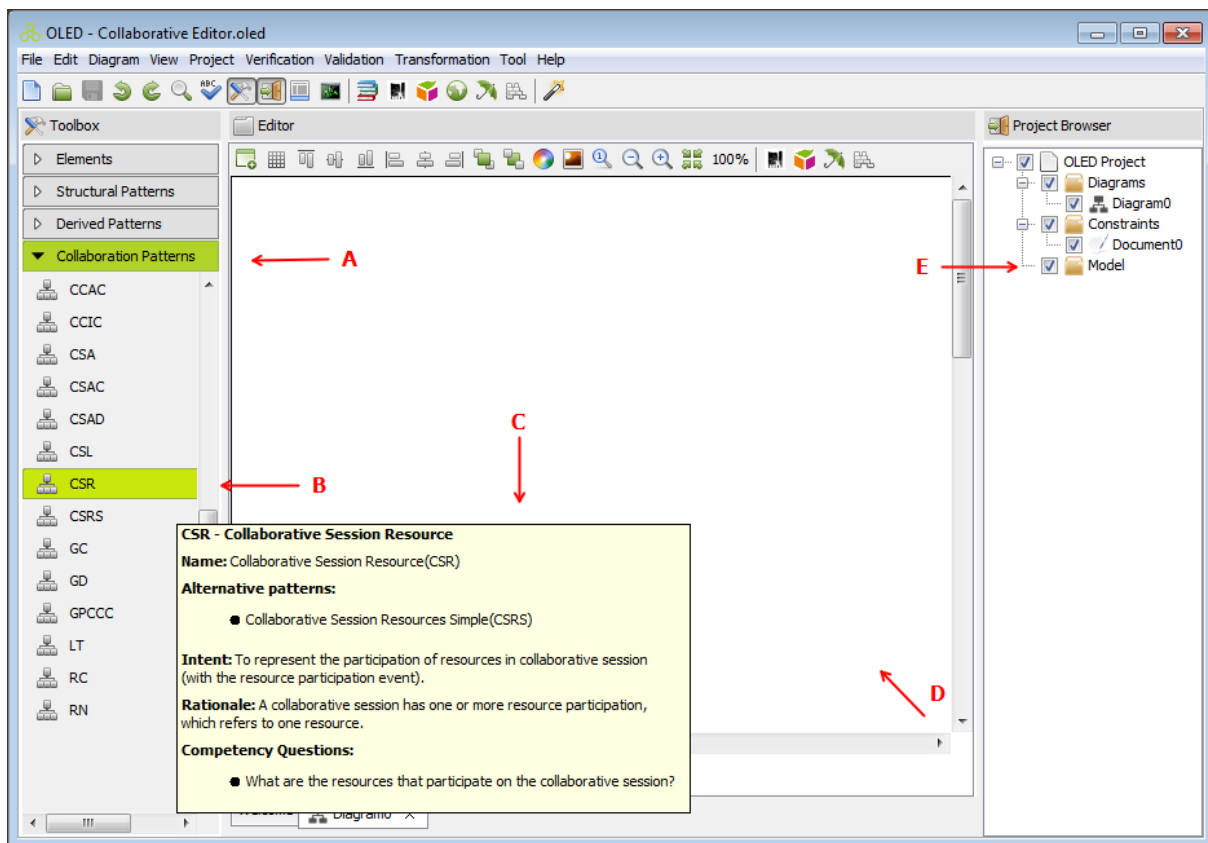


Figura 57 – Seleção do padrão CSR do catálogo

Quando na área do diagrama é clicada com o *mouse*, após a seleção do OP do catálogo, será aberta uma tela para configurações de geração do OP com uma pré-visualização do OP e uma tabela com as classes pertencentes ao OP bem como algumas configurações possíveis para a geração. Na Figura 58 pode-se ver a tela com seus elementos em que podem ser vistos alguns deles em destaque e que tem as seguintes funções:

- (A): Coluna *Disable class?* com botões *checkbox* para desabilitar a geração da classe correspondente;
- (B): Coluna *Class Name* para os nomes das classes pertencentes ao OP (*CSR*);
- (C): Coluna *Stereotype* para os estereótipos das classes pertencentes ao OP (*CSR*);
- (D): Coluna *Pick up from model?* em que é possível habilitar a reutilização da outra classe pertencente ao projeto no momento da geração do OP;
- (E): Botão de acesso a informações de ajuda sobre as categorias OntoUML das quais as classes do OP pertencem;
- (F): Tabela de configuração da geração do OP selecionado;
- (G): Botão *Create classes* responsável pelo disparo da ação de geração do OP selecionado (*CSR*) no diagrama corrente;
- (H): Painel de pré-visualização do OP selecionado para utilização;
- (I): Botão *Show Image* responsável por abrir uma tela para pré-visualização do OP selecionado (*CSR*);

Se o modelador julgar necessário, ele pode ter acesso a uma tela de ajuda sobre as categorias OntoUML pertencentes ao diagrama clicando no botão mostrado na Figura 58 (E). Dessa forma, ele verá a tela apresentada na Figura 59. Nessa figura, é apresentada a tela com elementos em destaque alguns deles e que tem as seguintes funções:

- (A): Listagem com as categorias OntoUML pertencentes ao OP selecionado (*CSR*);
- (B): Botão *Open OntoUML Tree* que, quando acionado, abre uma tela com a árvore de categorias OntoUML;
- (C): Botão *Close* que, quando acionado, fecha a janela de ajuda;
- (D): Painel com algumas informações sobre a categoria OntoUML selecionada (A);

Caso o modelador tenha interesse em visualizar a árvore de categorias OntoUML, Figura 59 (B), será mostrada uma tela com a árvore. Na Figura 60 pode ser vista a tela que tem os seguintes elementos com suas respectivas funções:

- (A): Painel com a árvore das categorias OntoUML;
- (B): Botão *Close* que, quando acionado, fecha a tela da árvore de categorias OntoUML;

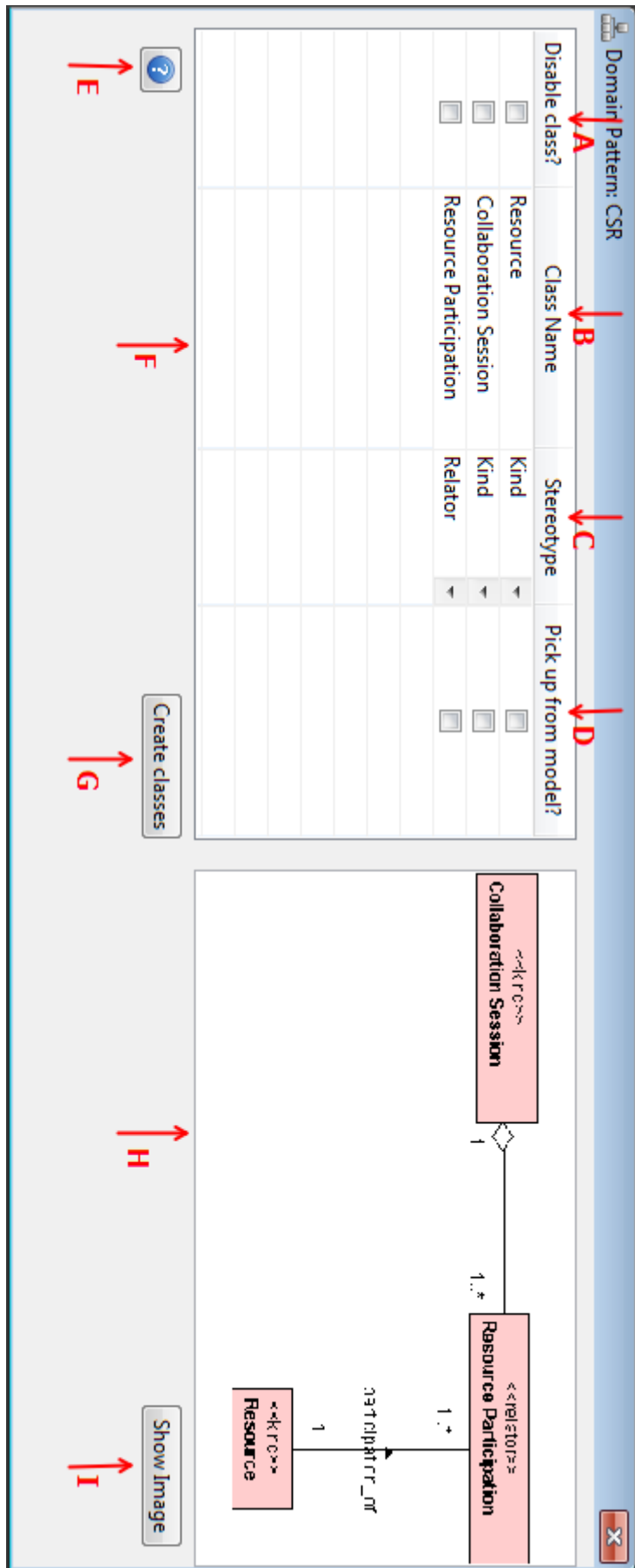


Figura 58 – Características do padrão CSR do catálogo

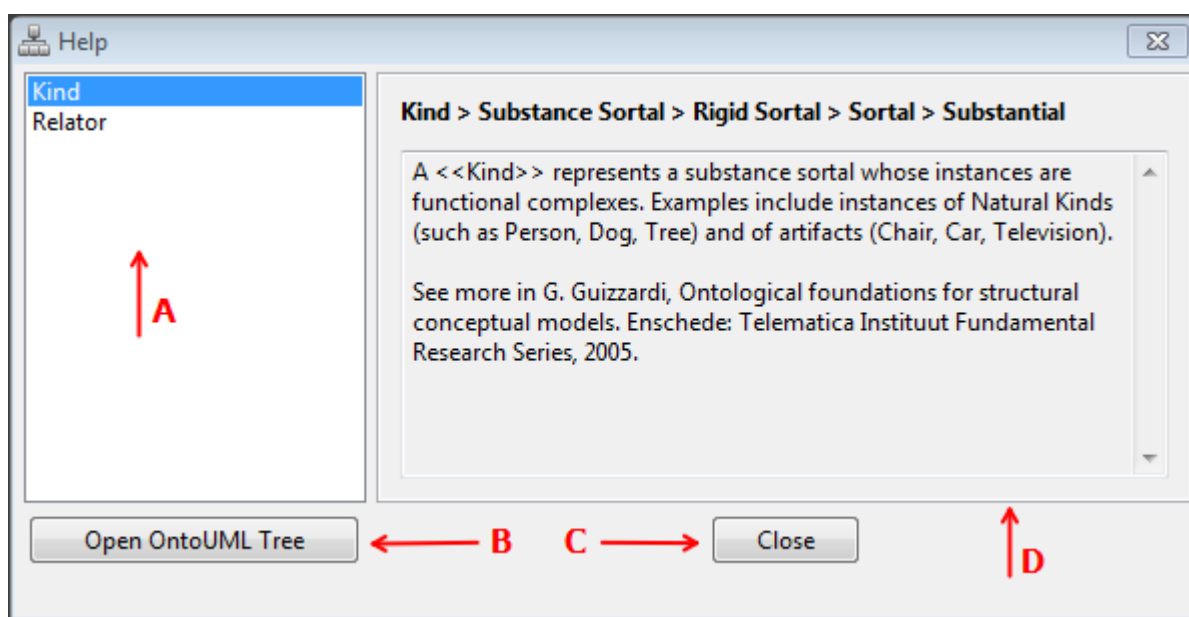


Figura 59 – Ajuda sobre as categorias OntoUML do OP

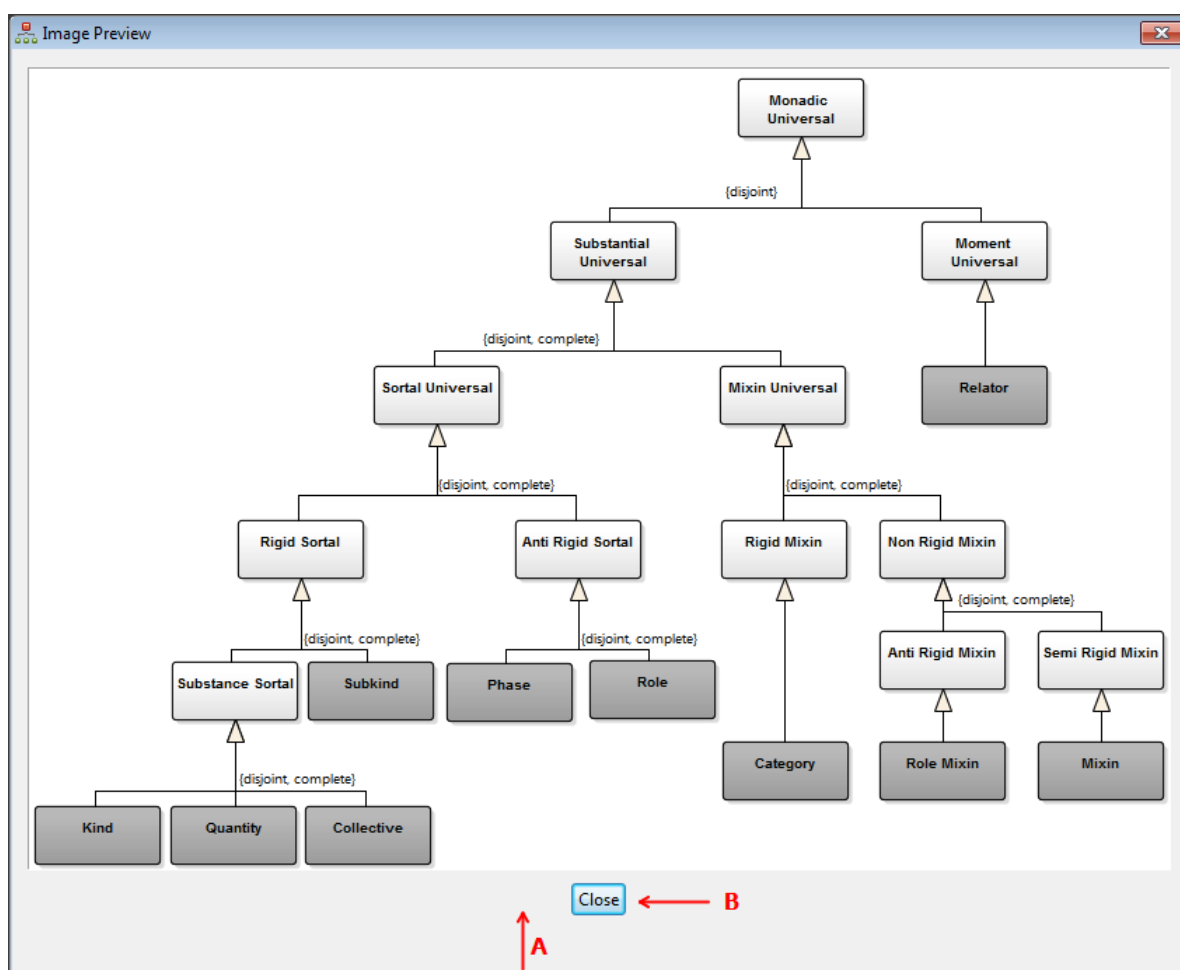


Figura 60 – Ávore de categorias de OntoUML

Quando fechada tela, Figura 60 e retornado para a tela da Figura 58, acionado o botão *Create classes* (*G*), o OP será criado no diagrama conforme Figura 61.

Na Figura 61, o OP *CSR* é desenhado no diagrama e seleção do OP *RN*. O processo para utilização do novo OP (*RN*) é o mesmo apresentado anteriormente, mas, desta vez, será configurada a geração para que seja reutilizado a classe *Resource* do OP *CSR* já presente no projeto. Dessa forma, ainda podem ser vistos alguns elementos destacados na Figura 61 que tem as seguintes funções:

- (A): Entrada na caixa de ferramentas (*toolbox*) do catálogo de DROPs, apresentando o nome do catálogo que foi definido na tela apresentada na Figura 48 (*B*);
- (B): OP (*RN*), pertencente ao catálogo de DROPs selecionado, para utilização na construção de uma ontologia operacional no domínio colaborativo;
- (C): Balão mostrando as informações resumidas do OP *RN* que foram configuradas nas telas apresentadas nas Figuras 48, 50, 51, 52 e 53;
- (D): Área de trabalho representando o conteúdo do diagrama corrente em que é desenhada a ontologia, ou seja, local em que deve ser clicado com o *mouse*, após a seleção do OP no catálogo (*B*), para a criação do OP;
- (E): OP (*CSR*), pertencente ao catálogo de DROPs selecionado, gerado para a construção de uma ontologia operacional no domínio de colaboração;
- (F): Nó *Model* da árvore de navegação do projeto em que são adicionadas as classes OntoUML pertencentes ao projeto. As classes OntoUML pertencentes aos OPs do catálogo serão adicionadas nesse nó quando esses OPs forem utilizados no projeto no projeto;
- (G): Classes OntoUML pertencentes ao OP *CSR* gerado no diagrama (*E*);

Já na Figura 62, algumas informações de configurações possíveis para se fazer na geração das classes do OP *RN* podem ser vistas além dos elementos participantes dessas configurações que estão destacados e que tem as seguintes funções:

- (A): Coluna *Disable class?* com botões *checkbox* para desabilitar a geração da classe correspondente;
- (B): Coluna *Class Name* para os nomes das classes pertencentes ao OP;
- (C): Coluna *Stereotype* para os estereótipos das classes pertencentes ao OP;
- (D): Coluna *Pick up from model?* em que é possível habilitar a reutilização de alguma outra classe pertencente ao projeto, marcando o *checkbox* (*E*), o que fará com que a classe selecionada seja reaproveitada para a geração do OP;

The screenshot displays the Collaborative Editor interface. At the top, the menu bar includes File, Edit, Diagram, View, Project, Verification, Validation, Transformation, Tool, and Help. The toolbar contains various editing tools. The Project Browser on the left shows a tree structure with folders for OLEO Project, Diagrams, Constraints, Document0, and Model. The main editor area shows an ontology diagram with three classes: Collaboration Session (labeled with <<kind>>), Resource Participation (labeled with <<relator/>>), and Resource (labeled with <<kind>>). Relationships include a composition (1 to 1..*), a generalization (1 to 1..*), and an association (1 to 1..*). A pop-up window for the 'RN - Resource Nature' pattern is visible, containing the following information:

- Name:** Resource Nature(RN)
- Alternative patterns:** --
- Intent:** To represent resource types in terms of shareability.
- Rationale:** A resource can be either sharable or exclusive.
- Competency Questions:**
 - Can this resource be shared?

Red arrows labeled A through G point to specific elements in the interface: A points to the 'RN' pattern in the catalog; B points to the pop-up window; C points to the 'Resource' class; D points to the 'participation_of' relationship; E points to the 'Collaboration Session' class; F points to the 'Resource Participation' class; and G points to the 'Resource Participation' class in the Project Browser.

Figura 61 – Seleção do padrão RN do catálogo

- (E): Botão *checkbox* de confirmação de reutilização de outra classe no lugar da classe do respectivo registro;
- (F): Botão de acesso a informações de ajuda sobre das categorias OntoUML as quais as classes do OP pertencem;
- (G): Tabela de configuração da geração do OP selecionado;
- (H): Botão *Create classes* responsável pelo disparo da ação de geração do OP selecionado no diagrama corrente;
- (I): Painel de pré-visualização do OP selecionado para utilização;
- (J): Botão *Show Image* responsável por abrir uma tela para pré-visualização do OP selecionado;

A Figura 63 apresenta a pré-visualização do OP (*RN*) com as classes e as relações que o compõem e também os elementos participantes da pré-visualização do OP selecionado, bem como alguns elementos destacados que têm as seguintes funções:

- (A): Botão *Close* responsável para fechar a tela de pré-visualização do OP;
- (B): Painel de visualização do OP selecionado para utilização;

Já na Figura 64, é apresentada uma árvore com as classes presentes no diagrama agrupadas pelas categorias OntoUML para que o modelador tenha a opção de selecionar alguma delas para reuso, ou seja, para que não seja gerada uma nova classe no diagrama, uma vez que a classe de interesse já existe no diagrama. Na figura ainda se têm os elementos participantes da seleção da classe para reuso do OP selecionado. Nela pode-se ver destacados alguns elementos que tem as seguintes funções:

- (A): Categoria OntoUML que é utilizada para agrupar as classes presentes no diagrama;
- (B): Classes presentes no diagrama disponíveis para reuso para o OP escolhido;
- (C): Campo de entrada disponível para informa o nome da classe que se deseja reutilizar para que seja procurada na árvore de classes;
- (D): Botão *Search* responsável pelo disparo da ação de busca da classe informada em *C*;
- (E): Botão *Select* de confirmação de seleção da classe para reuso *B*;

Por último, é mostrado na Figura 65 os OPs *CSR* e *RN* gerados no diagrama com a reutilização da classe *Resource*. Também estão destacados outros elementos da tela apresentada na mesma figura que tem as seguintes funções:

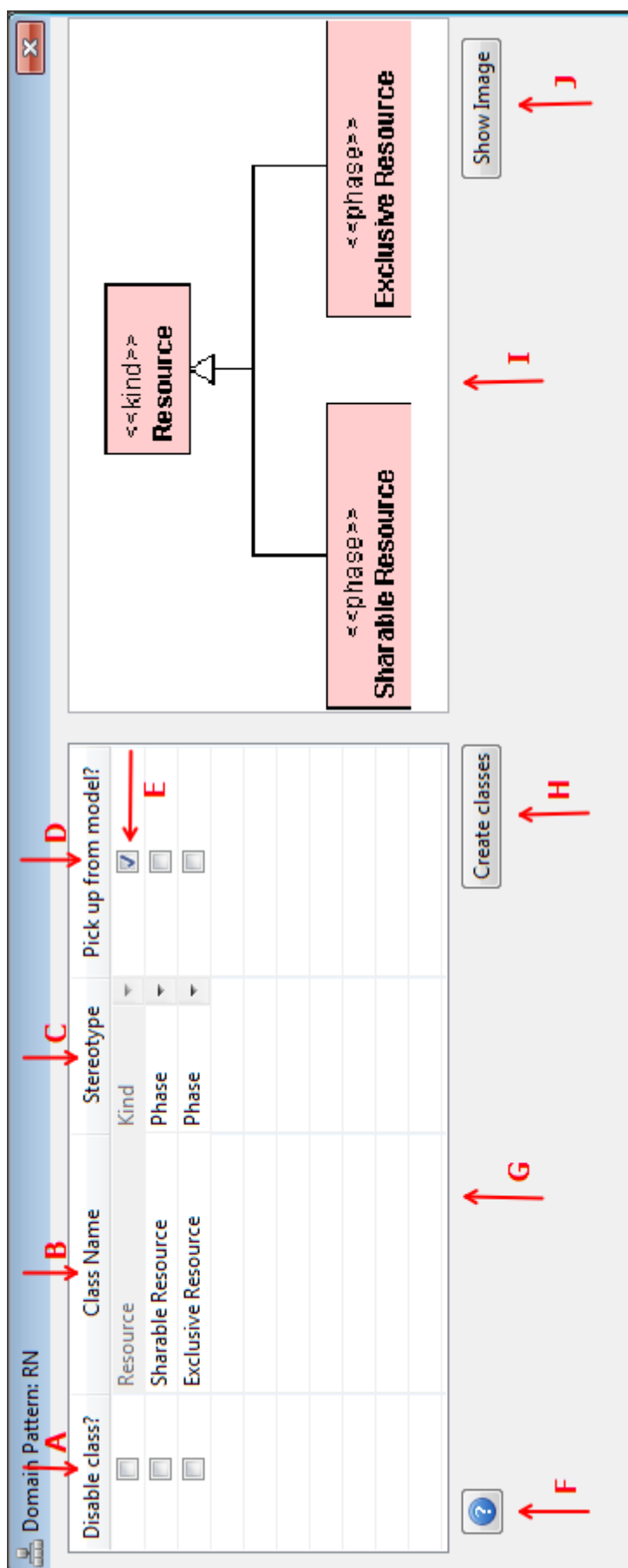


Figura 62 – Características do padrão RN do catálogo

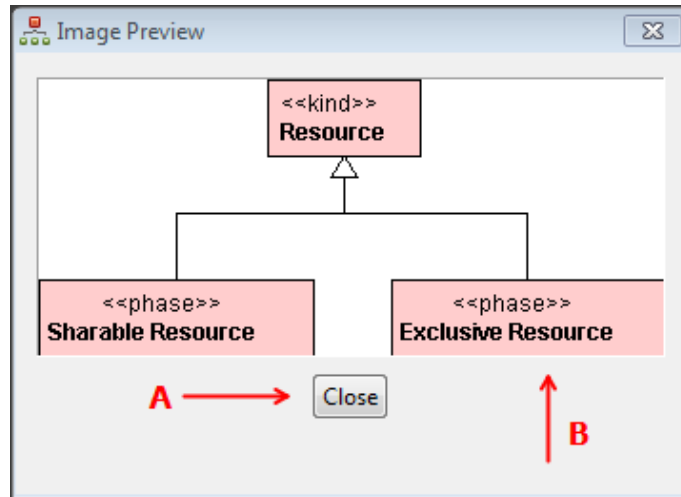


Figura 63 – Pré-visualização do OP RN

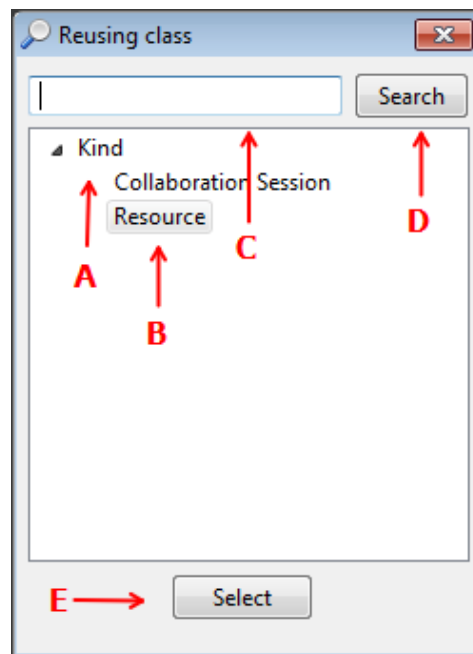


Figura 64 – Réuso da classe *Resource* do OP

- (A): OP (*CSR*), pertencente ao catálogo, gerado para a construção de uma ontologia operacional no domínio de colaboração;
- (B): OP (*RN*), pertencente ao catálogo, gerado para a construção de uma ontologia operacional no domínio de colaboração reutilizando a classe OntoUML *Resource*;
- (C): Nó *Model* da árvore de navegação do projeto em que são adicionadas as classes OntoUML pertencentes ao projeto. Pode-se perceber que não existem duas entradas para a classe *Resource* uma vez que ela foi reutilizada na geração do OP *RN*;
- (D): Classes OntoUML pertencentes aos OP *CSR* e *RN* gerado no diagrama (*B* e *C*);

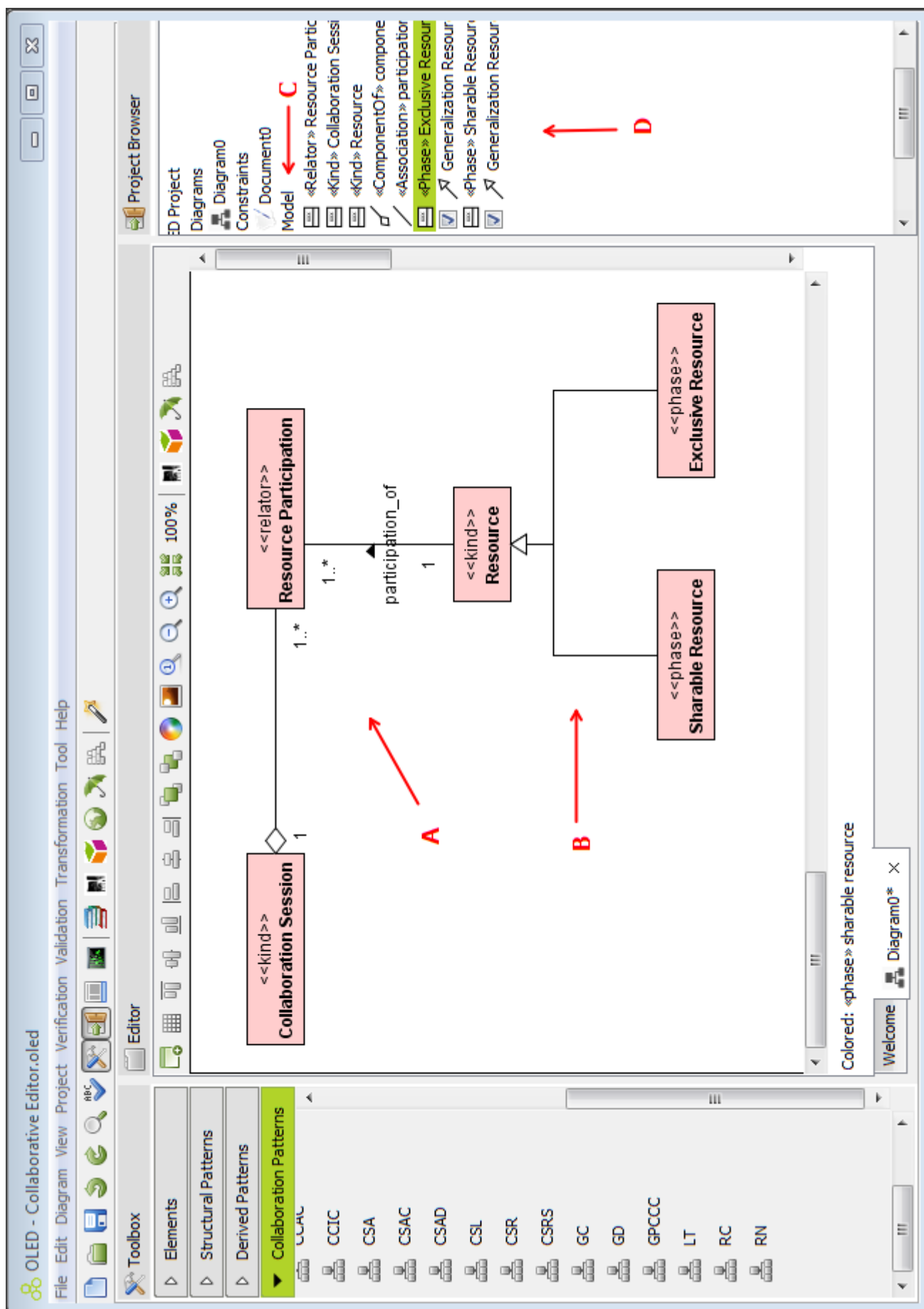


Figura 65 – Ontologia parcial com os OPs CSR e RN

6 Exemplos de Uso de Padrões Ontológicos para Construir Ontologias Operacionais

Conforme [Quirino et al. \(2015\)](#)), ontologias têm sido reconhecidas como um instrumento muito importante para reduzir ambiguidades conceituais e inconsistências. As soluções ontológicas estão ganhando cada vez mais importância nas atividades que exigem uma compreensão mais profunda em algum domínio. Com isso, as ontologias, como modelos de referência, podem ser úteis para compartilhar a conceituação pertinente a esses domínios, estruturando-os para que sejam comunicados adequadamente com os interessados, por exemplo, entre departamentos de uma empresa e seus respectivos parceiros

Mas como ressaltado nos capítulos [1](#) e [2](#) desta dissertação, a construção de ontologias não é uma tarefa fácil e exige conhecimento e esforço. Assim, a reutilização é atualmente reconhecida como uma prática importante para a Engenharia de Ontologia, e as abordagens orientadas por padrões são promissoras para apoiar a reutilização da ontologia ([BUSCHMANN; HENNEY; SCHMIDT, 2007](#)), uma vez que um padrão ontológico descreve um problema de modelagem particular recorrente que surge em contextos de desenvolvimento específico e apresenta uma solução bem comprovada ([FALBO et al., 2013b](#)).

Este capítulo descreve três exemplos do uso da abordagem de catálogos de DROPs proposta nesta dissertação (capítulo [4](#)), que é apoiada pelo editor descrito no capítulo [5](#). Dessa forma, este capítulo serve como validação do presente trabalho, explorando e verificando a viabilidade do que aqui foi proposto e desenvolvido. Para isso, buscou-se casos que dessem margem a discussões distintas em relação ao suporte aos Engenheiros de Ontologias na construção de ontologias operacionais a partir do catálogo automatizado de DROPs. Isso se dá a partir de estudos de caso tomando por base os domínios e padrões apresentados no capítulo [3](#): o domínio (e, conseqüentemente, os padrões) ligados a Serviços (seção [6.1](#)), aquele ligado a Processos de Software (seção [6.2](#)) e, finalmente, os relacionados a Colaboração (seção [6.3](#)).

6.1 Domínio de Serviço

Conforme exposto na seção [3.1](#), serviço é uma das áreas da economia que mais cresce. Muitos modelos empresariais de negócio são baseados em serviços. Essas empresas precisam representar e compartilhar conhecimento sobre seus modelos de serviço e as ontologias são instrumentos úteis para a representação e compartilhamento do conhecimento. Essa característica é de fundamental importância para empresas que querem prover serviços

com cada vez mais qualidade, integrar esses serviços a outros e se destacarem no mercado (FALBO et al., 2016).

O uso de ontologias para a modelagem dos serviços prestados por uma organização se torna peça chave para que eles sejam realizados de forma a agregar valor tanto para os provedores quanto para os consumidores de serviços. Dessa forma, utilizando os catálogos de DROPs de serviço apresentado de forma parcial na seção 3.1, será explorada a aplicabilidade da abordagem de catálogo de DROPs e do suporte automatizado proposto como uma ferramenta de reúso de ontologias por meio de OPs como suporte a construção de ontologias.

Alguns trabalhos já foram realizados provendo mecanismos para construção de ontologias baseado em OPs no domínio de Serviços. Um deles é o que é apresentado por Quirino et al. (2015) com a construção de uma subontologia de oferta de serviço, no contexto da utilização da S-OPL, uma linguagem de padrões ontológicos de serviço. Para a construção desta subontologia por meio da seleção dos OP, o critério adotado foi a elicitación das questões de competência que serviram para determinar quais OPs seriam utilizados. As questões de competência e os OPs escolhidos são apresentados na Tabela 27. Com os OP identificados, o resultado da modelagem pode ser visto na Figura 66.

Tabela 27 – Subontologia de oferta de serviço - mapeamento entre as questões de competência e os padrões

Identificador	Questão	Padrão
CQ01	Quais são os tipos de prestadores de serviços?	O-Provider
CQ02	Quais são os tipos de clientes-alvo?	O-OU-Customer
CQ03	Qual é a oferta de serviços estabelecida?	S-Offering
CQ04	Quais são os membros da comunidade alvo?	TCCMembership

Apesar das questões de competência ajudarem na escolha de OPs, o mapeamento entre os requisitos de modelagem e as questões de competência mapeadas para os OPs, dificulta o trabalho de modelagem pois requer um certo grau de conhecimento do domínio para que não haja problemas. Isso não quer dizer que é impossível usar esse método, mas que existem riscos a modelagem ao utilizá-lo, por exemplo, devido a inexperiência no domínio, pode resultar (i) na escolha de OPs inadequados; (ii) num processo de escolha lento; (iii) num retrabalho em uma modelagem e (iv) numa modelagem imprecisa, ficando muita responsabilidade a cargo do engenheiro de ontologias para mapear entre os requisitos de modelagem do domínio e as questões de competência.

Por ser um processo manual, sem apoio de ferramenta computacional, todo o trabalho se dá a partir no nível mais baixo de modelagem, ou seja, pelas classes OntoUML que compõem o modelo. Dessa forma, os itens (i), (ii), (iii), (iv) ficam mais potencializados. Dessa forma, abre-se um precedente devido a subjetividade para inconsistência da modelagem. Outro ponto deve-se a falta de uma estratégia mais produtiva para a escolha dos OPs para que o processo de seleção dos mesmos seja mais assertiva, eficaz e

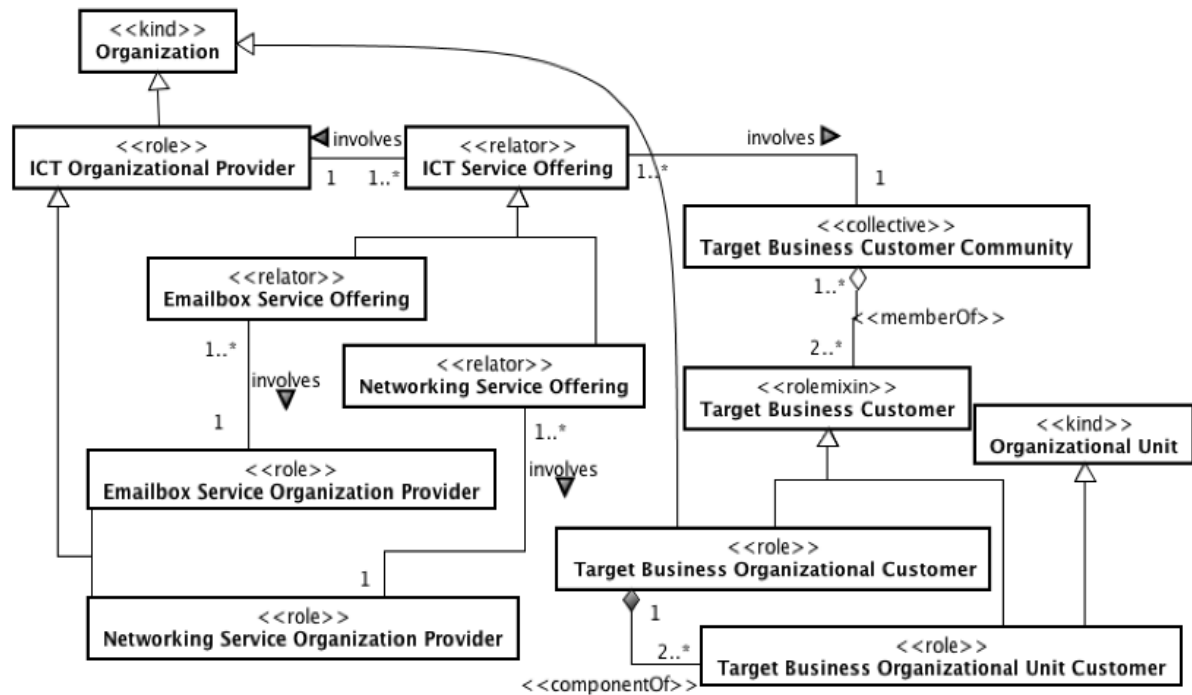


Figura 66 – Subontologia de oferta de serviço (QUIRINO et al., 2015)

rápida. Assim foi percebida a oportunidade de colaborar com o processo de modelagem de ontologias operacionais de domínio provendo uma ferramenta que dê suporte a trabalhar com a abordagem de catálogo de DROPs, o OLED estendido.

O OLED estendido, por ser uma ferramenta computacional, fornece o apoio necessário para a criação e utilização dos OPs da S-OPL organizados na forma de catálogo de DROPs com a anexação do processo de escolha dos OPs da S-OPL junto ao catálogo. Dessa forma os engenheiros de ontologias terão, não só uma coleção de DROPs organizadas em um catálogo, mas também terão um processo criado para orientar a escolha dos DROPs, junto do catálogo na própria ferramenta, bastando apenas que o criador do catálogo o tenha anexado esse processo. Na Figura 67, tem-se o ponto de acesso a visualização do processo anexado ao catálogo de DROPs de serviço. Já na Figura 68, é mostrado o processo de escolha dos DROPs do catálogo de serviço.

De posse catálogo de DROPs de serviços, ele será utilizado para realizar a mesma modelagem feita a partir das questões de competência representada pela Tabela 27 e Figura 66. O primeiro ponto a se observar é que as questões de competência têm um papel secundário, já que a escolha dos OPs para a modelagem se dará pelo processo de encadeamento dos DROPs do catálogo de serviço. Elas agora ajudam ao engenheiro de ontologias a entender quais problemas de modelagem o DROP trabalha. Os riscos a modelagem provenientes da inexperiência no domínio citados anteriormente, (i), (ii), (iii) e (iv) têm os respectivos efeitos anulados pela presença do processo de escolha dos DROPs presente no catálogo, acessível pelo OLED estendido que dá suporte computacional a

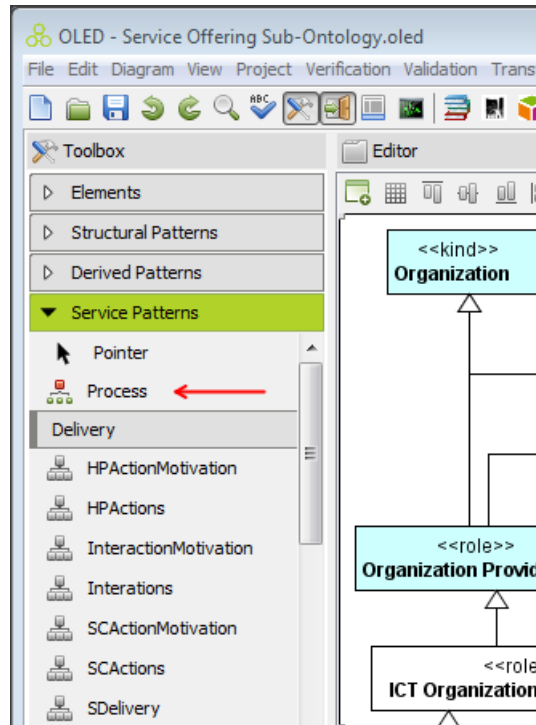


Figura 67 – Acesso ao modelo do processo da S-OPL

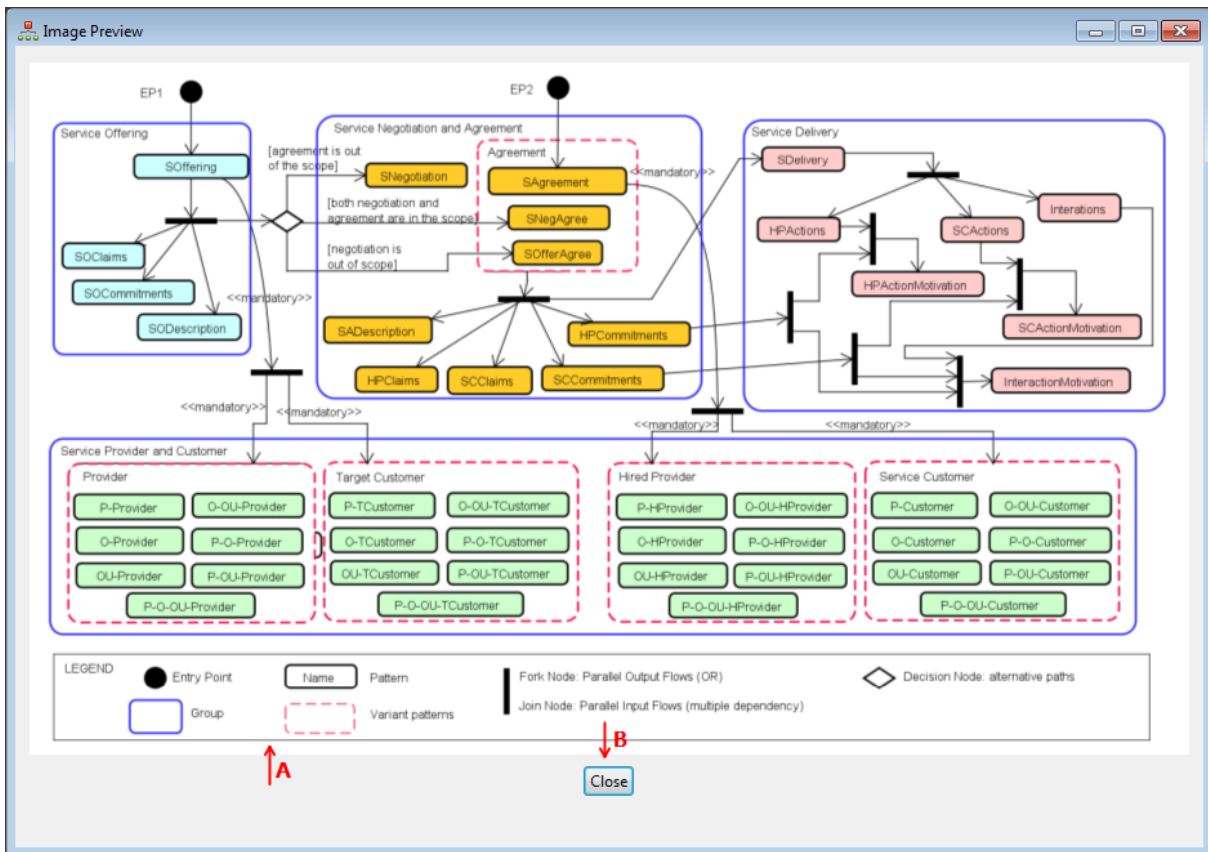


Figura 68 – Visualização do modelo do processo da S-OPL

modelagem tornando a construção da ontologia apresentada na Figura 66 mais produtiva. A seguir, será desenvolvida a mesma ontologia, porém utilizando o OLED estendido com o catálogo de DROPs de serviço.

Sendo assim, o primeiro ponto a se observar, é o que se deseja representar do domínio de serviço. No caso, uma subontologia de oferta de serviço. O modelo do processo da S-OPL prevê dois pontos de entrada para a seleção dos DROPs: EP1 e EP2 em que EP1 se dá pela oferta de serviço e EP2, pela negociação e acordos de serviço. Logo, o nosso ponto de entrada será EP1 porque é por onde se dá a oferta de serviço, interesse para a modelagem da ontologia. Daí tem-se o acesso aos OPs *S-Offering*, *O-Provider* e *O-OU-Customer* que foram utilizados na ontologia original. O OP *TCCMembership* não é apresentado porque na versão da S-OPL utilizada, este OP foi incorporado pelo *S-Offering*.

O processo de modelagem da ontologia da Figura 66 a Caso tivesse sido usado o OLED estendido, o processo de seleção dos OPs do catálogo teriam sido regidos pelo processo de seleção fornecido junto com a definição do catálogo de DROPs, como na Figura 68. Nele, têm-se os pontos de entrada (EP1 e EP2) que podem ser escolhidos para iniciar a escolha dos OPs e todo o encadeamento entre os OPs. Então, diante do modelo do processo da S-OPL apresentado, o ponto de entrada utilizado para modelar a ontologia na Figura 66 foi EP1 e os OPs escolhidos foram *S-Offering*, *O-Provider* e *O-OU-Customer*. O OP *TCCMembership* não é apresentado porque na versão da S-OPL utilizada, este OP foi incorporado pelo *S-Offering*. Estes DROPs foram apresentados seção 3.1.1.

Conforme o processo de utilização de um catálogo de DROPs apresentado no capítulo 5.2, estes DROPs foram adicionados ao modelo e estendidos para representarem os conceitos de oferta de serviço desejados. No processo de construção de ontologias operacionais de domínio é muito interessante notar que o mesmo se dá a partir de OP previamente modelados para lidar com problemas recorrentes e específicos nos respectivos domínios. É como se fosse adicionado uma camada entre os elementos mais baixos da linguagem de modelagem, classes OntoUML, e a ontologia operacional. Com o apoio computacional, OLED estendido, isso fica muito mais evidente e produtivo pois o OLED estendido adiciona a modelagem todas as classes do DROP e ainda possibilitando o reaproveitamento de classes já existentes no modelo, reúso. Com isso a produtividade aumenta bastante e o modelo se torna mais consistente. Na Figura 69 pode-se ver a ontologia resultante da modelagem utilizando o catálogo de DROPs de serviço com o suporte do OLED estendido.

Comparando as duas subontologias, sem o OLED estendido e com o OLED estendido, é perceptível o ganho de produtividade que se tem com a utilização da ferramenta. Por propiciar a criação de um catálogo de DROPs de serviço com um processo de encadeamento dos DROPs, na utilização do catálogo, o processo de construção da espinha dorsal da subontologia, os DROPs escolhidos do catálogo, se tornou muito prático, eficiente e

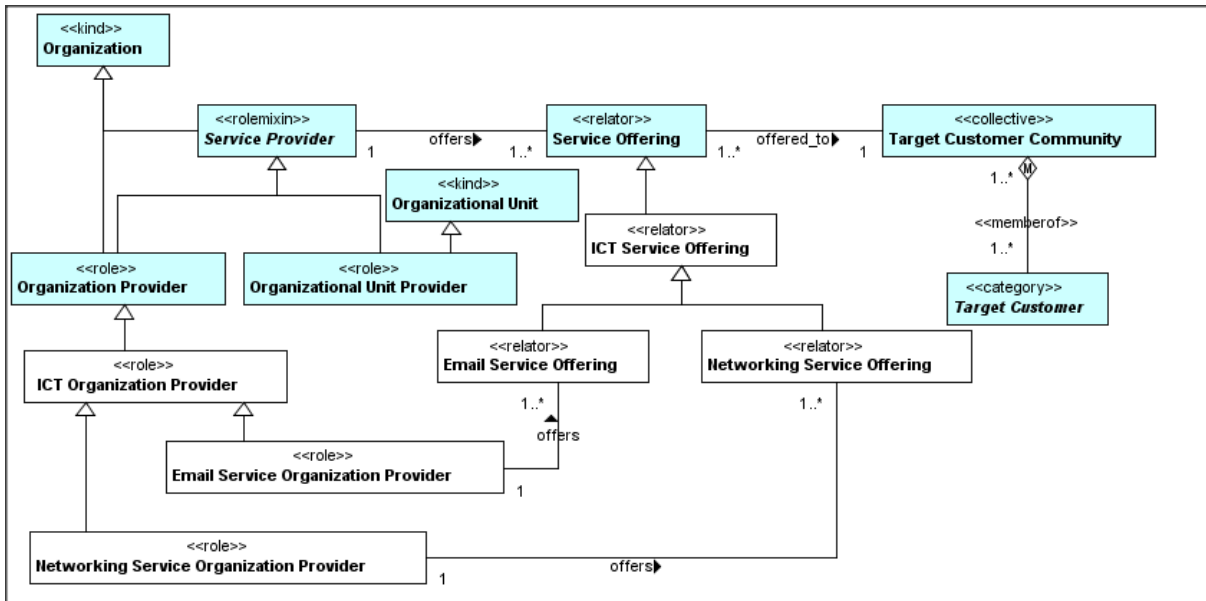


Figura 69 – Subontologia de oferta de serviço utilizando o OLED estendido

eficaz. Comparando as duas modelagens, percebe-se que o resultado alcançado utilizando o OLED estendido foi o mesmo, ou seja, se chegou a mesma ontologia a menos de diferenças de versões entre as S-OPLs de onde foram extraídos os DROPs para o catálogo.

Um outro exemplo é o da subontologia de acordo de serviço, apresentado por [Falbo et al. \(2016\)](#), também no contexto da utilização da S-OPL. Como já foi apresentado um exemplo de forma mais detalhada no domínio de serviço, este será mais direto. Para a construção desta subontologia por meio da seleção dos OPs, o critério adotado foi o processo da S-OPL. Com ele, foram selecionados os DROPs *SOffering*, *O-Provider*, *O-OUTCustomer*, *SOffering*, *SADescription*, *SCCommitments* e *HPCCommitments* resultando na modelagem apresentada na Figura 70.

Até aqui em nada é diferente da modelagem utilizando o OLED estendido. O que difere é que o OLED estendido fornece a processo, o catálogo e desenha o DROP escolhido no modelo com a possibilidade de reaproveitamento de conceitos previamente existentes. A S-OPL e o processo dela por si só já facilitam e ajudam muito o engenheiro de ontologias, mas com a organização do DROPs da S-OPL em catálogo e implementado em uma ferramenta computacional, OLED estendido, é o que torna a modelagem muito produtiva do ponto de vista ferramental.

Comparando as duas subontologias de acordo de serviço, com e sem a utilização do OLED estendido, é perceptível o ganho de produtividade que se tem com a utilização da ferramenta. Com o catálogo de DROPs de serviço e o modelo do processo para guiar a escolha dos DROPs no momento da utilização do catálogo, a base da subontologia, os DROPs escolhidos do catálogo, o trabalho se torna intuitivo e prático. Assim, fica perceptível que o resultado alcançado utilizando o OLED estendido foi o mesmo, ou seja,

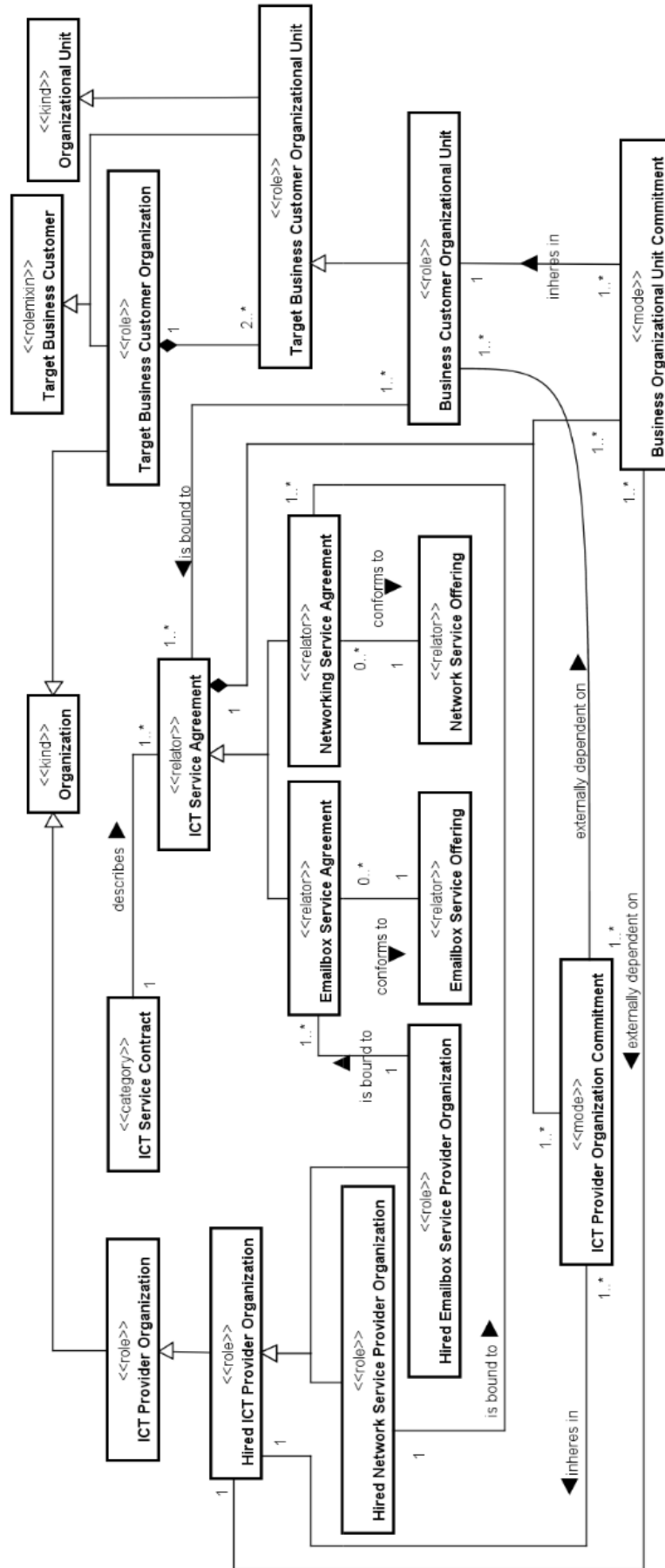


Figura 70 – Subontologia de acordo de serviço (FALBO et al., 2016)

se chegou ao mesmo resultado.

6.2 Domínio de Processo de Software

A Engenharia de Software (ES) foca nas diversas fases do processo de construção de uma solução computacional. Entre as atividades realizadas estão a especificação, desenvolvimento, manutenção e criação de sistemas de software que são realizadas com aplicação de tecnologias e práticas de gerência de projetos e outras disciplinas que têm por objetivo garantir organização, produtividade, qualidade e sistematização. Segundo [Falbo et al. \(2015b\)](#), um dos desafios permanentes em Engenharia de Software é lidar com questões de qualidade, visando entregar um produto de software resultante com maior produtividade e custos mais baixos.

Com isso, vários padrões de qualidade e modelos de maturidade, como ISO/IEC 12207, ISO/IEC 15504, CMMI, entre outros, têm sido utilizados para orientar os esforços das organizações de software para processos de software de qualidade ([RUY et al., 2015a](#)). Essas iniciativas buscam a melhoria do processo de software por meio da disseminação das melhores práticas de forma organizada e padronizada evitando inconsistências devido a utilização de diferentes padrões em conjunto, causados por problemas de interoperabilidade semântica. Nesse sentido, foram realizados esforços para criação de uma OPL de processo de software (ISP-OPL) para apoiar a criação de ontologias do domínio de processo de software ([FALBO et al., 2015b](#); [RUY et al., 2015a](#)).

As organizações de software têm investido cada vez mais na melhoria de seus processos de software. Estes, por sua vez, são processos intensivos de conhecimento que envolvem muitas pessoas que trabalham em diferentes subprocessos e atividades. Além disso, o conhecimento em SE é diverso e as organizações têm problemas para capturá-lo, recuperá-lo e reutilizá-lo ([RUY et al., 2016](#)). Assim o uso de ontologias para a modelagem de processo de software se torna cada vez mais interessante para que elas sejam realizadas de forma a agregar valor as organizações de software.

Neste sentido, alguns trabalhos vêm sendo realizados provendo mecanismos para construção de ontologias baseado em OPs no domínio de processo de software. Um deles é o que é apresentado por [Quirino et al. \(2015\)](#) onde é construída uma ontologia de requerimento do software, com utilização da ISP-OPL, uma linguagem de padrões ontológicos de processo de software. A escolha dos DROPs da ISP-OPL se deu pelo fluxo de relacionamentos dos DROPs definidos na mesma e que pode ser visto na Figura 71. O ponto de entrada escolhido foi o EP2 o que levou a seleção dos DROPs *PWUC*, *PWUD*, *PPP*, *StD*, *PPas*, *WPN*, *WPC*, *DocD*, *WPCrea*, *WPUse* e *WPChan*. Todos esses OPs escolhidos foram apresentados na seção 3.2.1 e o resultado da modelagem pode ser visto na Figura 72.

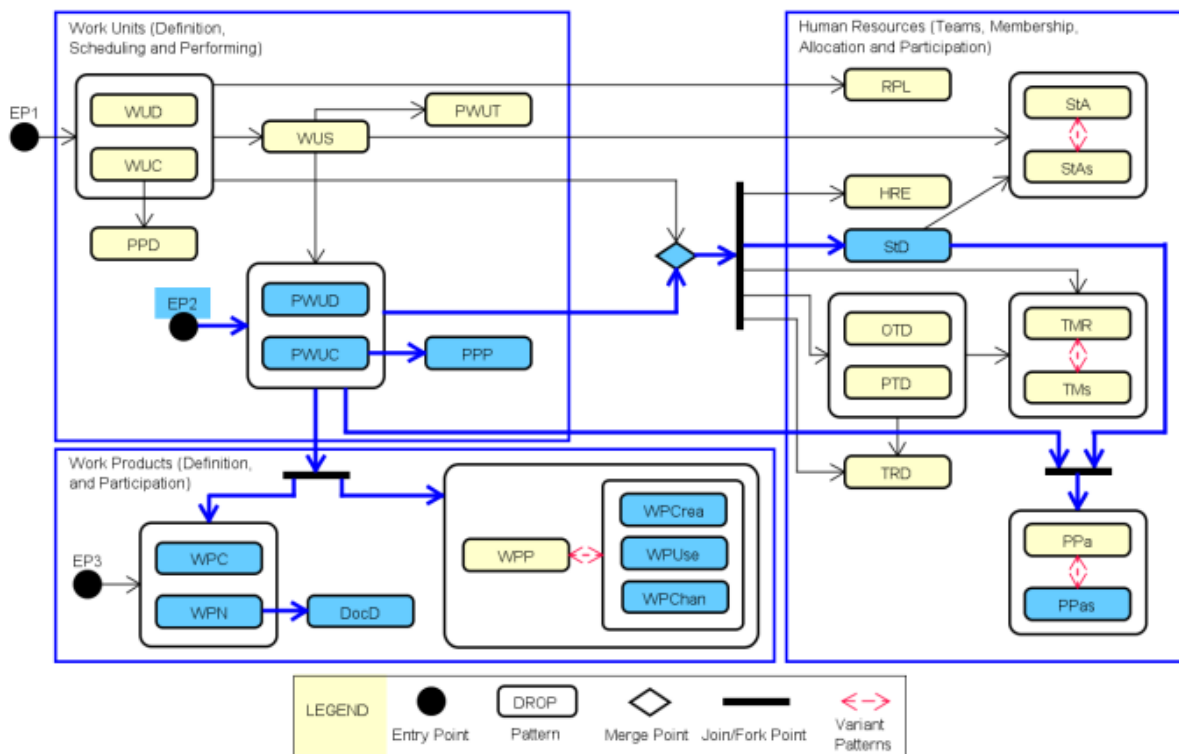


Figura 71 – Processo de software baseado na ISP-OPL (RUY et al., 2015b)

Como pode ser percebido, as OPLs fornecem um processo bem definido para a modelagem especificando tanto os OPs que os compõem quanto do relacionamento entre eles. Essa característica vem do fato de que, conforme Ruy et al. (2015b), uma OPL é uma rede de padrões de ontologia relacionados ao domínio interconectados que fornece suporte para resolver uma classe de problemas de desenvolvimento de ontologia para um domínio específico. Do ponto de vista de modelagem, OPLs são ferramentas muito robustas para a construção de ontologias operacionais. Mas ainda não são implementadas em sistemas computacionais, nem têm suas utilizações apoiadas por ferramentas automatizadas que implementem a abordagem de construção de ontologias baseadas em OPLs.

Vale ressaltar que, pelo fato de OPLs serem definições e especificações de DROPs inter-relacionados, sem um apoio computacional, até onde pode ser averiguado, alguns problemas podem surgir na sua utilização. Por exemplo (i) falta de ferramenta para desenhar as classes OntoUML que compõem o DROP; (ii) dificuldade de compreensão da modelagem devido a distribuição inadequada dos elementos; (iii) desenho, a partir do zero, das classes pertencentes aos DROPs selecionados da OPL, (iv) risco de perda de todo o trabalho devido a mudança de objetivo da modelagem devido a escolha de outros DROPs não utilizados e descarte de algum utilizado. O apoio computacional viria bem a calhar sanado, entre outros, os problemas citados.

O OLED estendido, mesmo não sendo uma ferramenta para trabalho com OPLs, proporciona um grande auxílio aos engenheiros de software pois implementa a abordagem

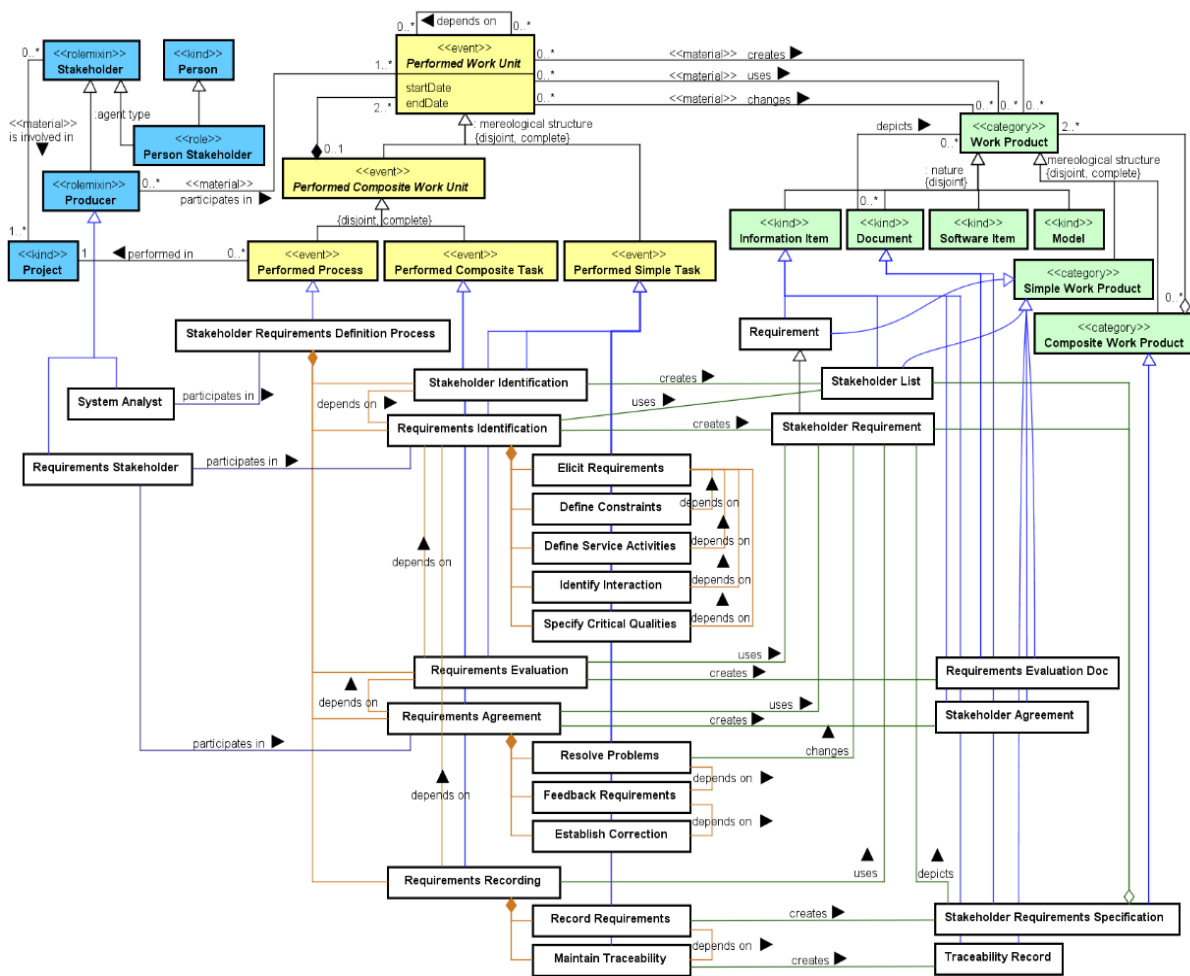


Figura 72 – Ontologia do processo de requisitos (sub-ontologia do processo de definição de requisitos das partes interessadas) (RUY et al., 2015a)

de catálogo de DROPs, possibilitando a criação e utilização dos mesmos na construção de ontologias com a anexação do processo da OPL. A fim de verificar os ganhos, será explorada a aplicabilidade da abordagem de catálogo de DROPs com suporte automatizado fornecido pelo OLED estendido como uma ferramenta de reuso de ontologias por meio de DROPs como suporte a construção de ontologias operacionais neste domínio, processo de software baseado na ISO.

O primeiro ganho que o OLED estendido proporciona é referente a criação de um catálogo de DROPs baseados na ISP-OPL. Assim os DROPs da ISP-OPL ficaram organizados, estruturados e implementados com o apoio de uma ferramenta computacional, o OLED estendido. Como as OPLs são uma rede de DROPs interconectados e a ISP-OPL tem uma especificação tanto dos DROPs quanto do processo de relacionamento dos mesmos, os catálogos fornecem uma estrutura organizacional em que os DROPs de uma OPLs podem ser empacotados e distribuídos. O processo de criação de catálogos de DROPs com o OLED estendido foi detalhado no capítulo 5 para maiores informações.

Com a especificação da ISP-OPL (FALBO et al., 2015b) em mãos, os DROPs dela

foram implementados utilizando as classes OntoUML presentes no OLED estendido. O catálogo resultante foi configurado e o processo da ISP-OPL foi anexado ao catálogo. Vale ressaltar que a implementação foi realizada de forma parcial uma vez que nosso foco está na abordagem de criação e utilização de catálogos de DROPs com suporte automatizado por ferramenta computacional e não a proposição do catálogo de DROPs da ISP-OPL.

Dessa forma, os engenheiros de ontologias terão não só uma coleção de DROPs organizados em um catálogo, mas também um processo criado para orientar a escolha dos DROPs, junto ao catálogo na própria ferramenta utilizada para a modelagem. Isso pode ser visto na Figura 73 em que o ponto de acesso a visualização do modelo do processo anexado ao catálogo de DROPs está sinalizado pela seta vermelha e, na Figura 74, pode ser visto o modelo do processo da ISP-OPL que orienta a escolha dos DROPs do catálogo.

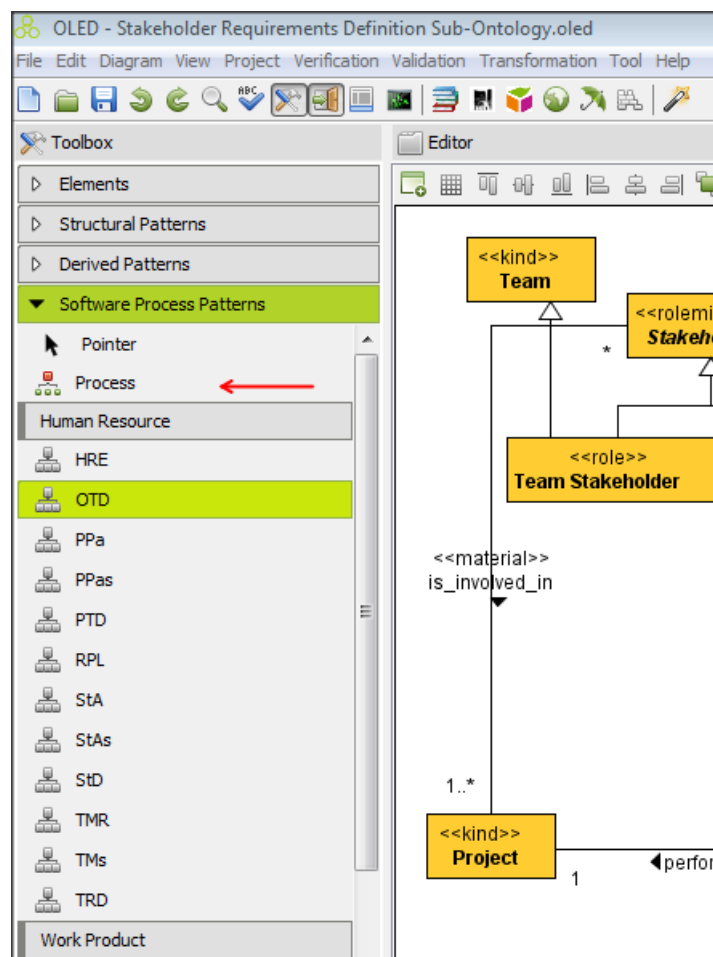


Figura 73 – Acesso ao modelo do processo da ISP-OPL anexado ao catálogo de DROPs

O fato de serem o catálogo de DROPs e o modelo do processo da ISP-OPL empacotados e distribuídos juntamente pelo OLED, faz com que a construção de uma ontologia operacional de processo de software se torne muito prático já que esta ferramenta suporta trabalhar com catálogos de DROPs na modelagem de ontologias, ou seja, uma ferramenta que já vem com um catálogo de DROPs no domínio desejado criado por

especialistas neste domínio e que também tem um processo para guiar o engenheiro de ontologias na escolha dos DROPs desejados. Caso não fosse a existência do OLED estendido, não se teria uma ferramenta computacional para apoiar o processo de modelagem e também não se teria a implementação do catálogo de DROPs. Assim, seria necessário um especialista no domínio para a realização da modelagem e ela seria feita a partir do zero com a utilização dos elementos de modelagem de nível mais baixo que seriam as classes OntoUML presentes na ferramenta.

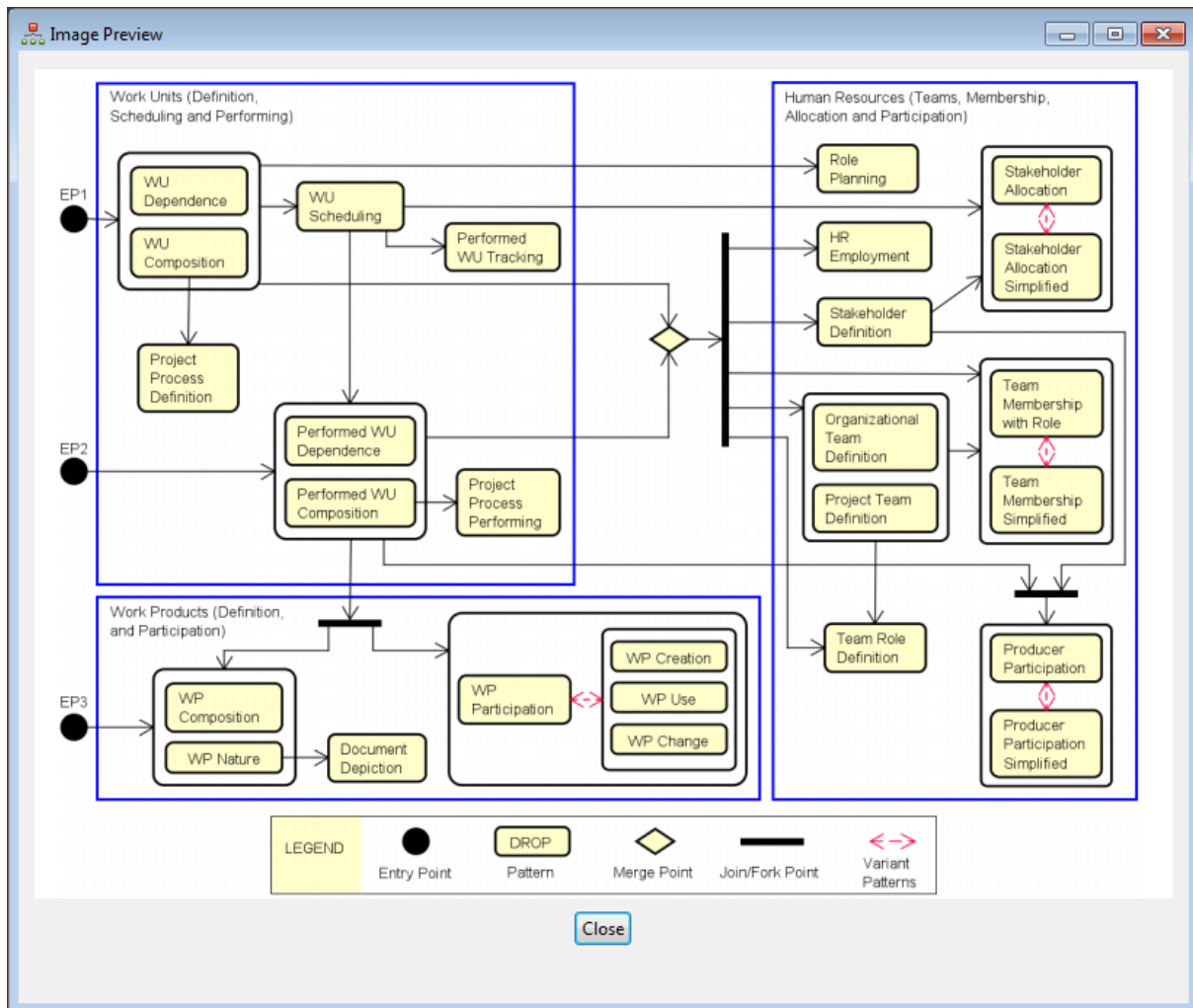


Figura 74 – Visualização do processo da ISP-OPL anexado ao catálogo de DROPs

Outro ponto de agilidade na construção de ontologias utilizando o OLED estendido tem relação com a produtividade e agilidade do desenho dos elementos de modelagem, as classes OntoUML. Como o OLED estendido trabalha com catálogos de DROPs, ao escolher o DROPs no catálogo presente na ferramenta e adicioná-lo a modelagem, a própria ferramenta se encarrega de desenhar e adicionar todas as classes OntoUML que o compõe na modelagem de forma automatizada, permitindo ainda que classes OntoUML presentes na modelagem sejam reutilizadas no DROP adicionado. Dessa forma, a abordagem de catálogo de DROPs acrescenta um nível de abstração entre os elementos de nível mais

baixo e a ontologia operacional resultante que é a camada de DROPs do catálogo.

Caso haja alguma alteração nos requisitos da modelagem e DROPs tenham que ser adicionados ou removidos dela, por estarmos utilizando o OLED estendido, podemos abrir o projeto da ontologia e remover e adicionar DROPs na modelagem, sem que haja descarte do modelo anterior. Sem o apoio computacional, isso não seria possível, e o modelo original seria descartado para a confecção de um novo que, muito provavelmente, seria criado do zero novamente.

Acessando o modelo do processo da OPL pelo mesmo ponto de entrada que foi utilizado (EP2), novamente os DROPs *PWUC*, *PWUD*, *PPP*, *StD*, *PPas*, *WPN*, *WPC*, *DocD*, *WPCrea*, *WPUse* e *WPChan* serão selecionados. Vale lembrar que todos esses DROPs foram apresentados na seção 3.2.1. Seguindo a seleção dos DROPs pelo modelo do processo presente no catálogo implementado no OLED estendido, chega-se a estrutura base de onde são estendidas as classes para a criação da ontologia operacional. Por questões de espaço, resolução da imagem com a modelagem e, assim, por acreditarmos não ser necessário desenhar a ontologia inteira, o resultado da modelagem somente com os DROPs escolhidos pode ser visto na Figura 75.

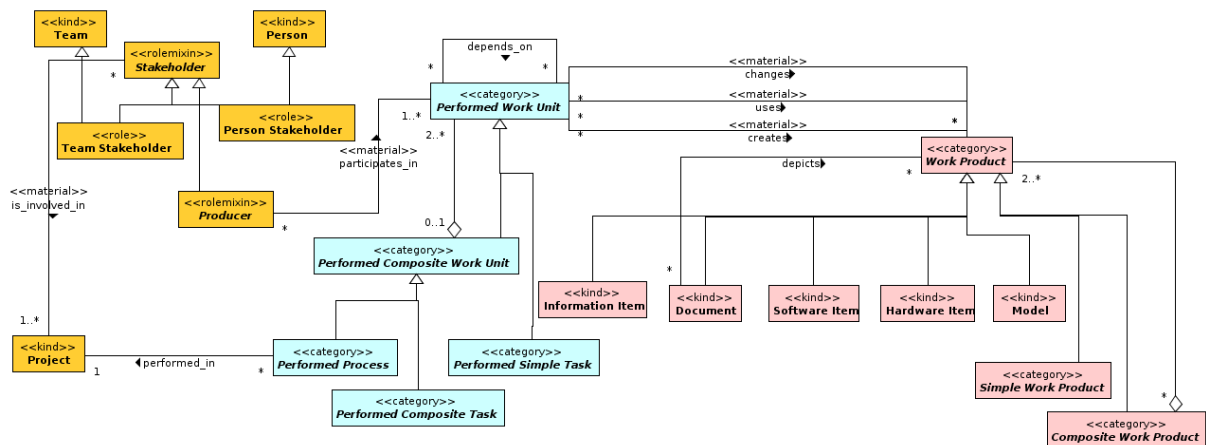


Figura 75 – Ontologia do processo de requisitos parcial (sub-ontologia do processo de definição de requisitos das partes interessadas)

Comparando as ontologias implementadas com e sem o apoio do OLED estendido, percebe-se que o resultado foi muito parecido, não sendo idêntico por uma questão de diferenças entre versões da ISP-OPL utilizadas, resultando em DROPs um pouco diferentes implementados no catálogo. Mas é perceptível a produtividade e praticidade que o OLED estendido trouxe ao processo de modelagem. Resultado que favorece a manutenibilidade do modelo e também do catálogo de DROPs. Havendo necessidade de alteração, basta abrir a modelagem e ajustar o que for necessário sem nenhum risco de perda do trabalho já realizado.

6.3 Domínio de Colaboração

A colaboração é uma das relações mais utilizadas na realização de atividades. Os desafios enfrentados por organizações e pessoas para construírem processos produtivos eficientes têm impulsionado a procura e criação de formas inovadoras para apoiar o crescimento. Isto decorre de muitos fatores tais como globalização das organizações e economias, serviços especialistas ofertados e consumidos, inovação constante da tecnologia e competitividade e disputa de mercado. Para a sobrevivência, parcerias, melhorias constantes dos processos e inovação tecnológica têm fomentado a colaboração nas organizações. Para serem pioneiras, as organizações utilizam a colaboração para combinar o potencial e a experiência de seus funcionários (KNOLL et al., 2010) para alcançar os objetivos.

Webster (2016) define colaboração como o trabalho realizado com outra pessoa ou grupo a fim de alcançar ou fazer algo. Ou seja, a contribuição se dá em conjunto a outros para se fazer algo que de outra forma não poderia ser feito ou seria feito com maior dificuldade. Terveen (1995), por sua vez, define colaboração como o processo de um grupo onde os participantes trabalham juntos para alcançar um objetivo compartilhado. A noção de compartilhamento se torna clara assim como o sentimento de comprometimento dos envolvidos. Gray (1989) expressa colaborações como um processo pelo qual as partes que veem diferentes aspectos de um problema exploram construtivamente suas diferenças e buscam soluções que vão além da própria visão limitada do que é possível.

A colaboração é influenciada por vários fatores tais como grupo, cultura, tarefa, contexto, expertise e tecnologia utilizada. Knoll et al. (2010) argumenta que os comportamentos de grupo resultantes podem levar a diferentes efeitos que influenciam a eficiência da colaboração e com isso o processo de inovação. Para reconhecer e gerenciar esses efeitos, as organizações precisam de experiência em projeto e execução de processos de colaboração. Neste sentido, ontologias podem ser utilizadas para contribuir com o processo de colaboração. Oliveira (2009a) nos apresenta uma abordagem para modelar a colaboração propondo a utilização DROPs para apoiar os engenheiros de ontologias a construírem ontologias operacionais de colaboração, a CONTO.

Lá são apresentados a estrutura das ontologias de referência e definidos DROPs de colaboração, porém, não são apresentados organizados na forma de catálogo nem implementados em ferramenta computacional. A evolução e aprimoramento da CONTO é um trabalho em curso, um deles é empacotar os DROPs extraídos da CONTO em um catálogo (SOUZA et al., 2016). Um aspecto que será explorado com a aplicação da abordagem de catálogo de DROPs, com suporte automatizado fornecido pelo OLED estendido, para o domínio de colaboração é como esta ferramenta e a abordagem por ele implementada podem auxiliar no projeto de um catálogo de DROPs. Alguns DROPs de colaboração definidos na CONTO serão utilizados na criação do catálogo e para a construção de uma ontologia operacional de colaboração. Eles podem ser vistos na seção

3.3 e, mais detalhadamente, no apêndice A.

No processo de construção e evolução do catálogo de DROPs de colaboração, o OLED estendido fornece suporte para a extração de DROPs de uma ontologia de referência, neste caso a CONTO. Essa funcionalidade permite ao engenheiro do catálogo selecionar as classes OntoUML, a partir de uma ontologia de referência, que formam o DROP e a ferramenta dispõe de uma funcionalidade para criar de forma automática a representação do DROP. Na Figura 76 pode ser visto um exemplo do que foi dito, ou seja, da seleção das classes OntoUML que formam o DROP (seta A) e o item do *menu* que dispara a criação do DROP (setas B e C). Este processo foi descrito e detalhado na seção 4.

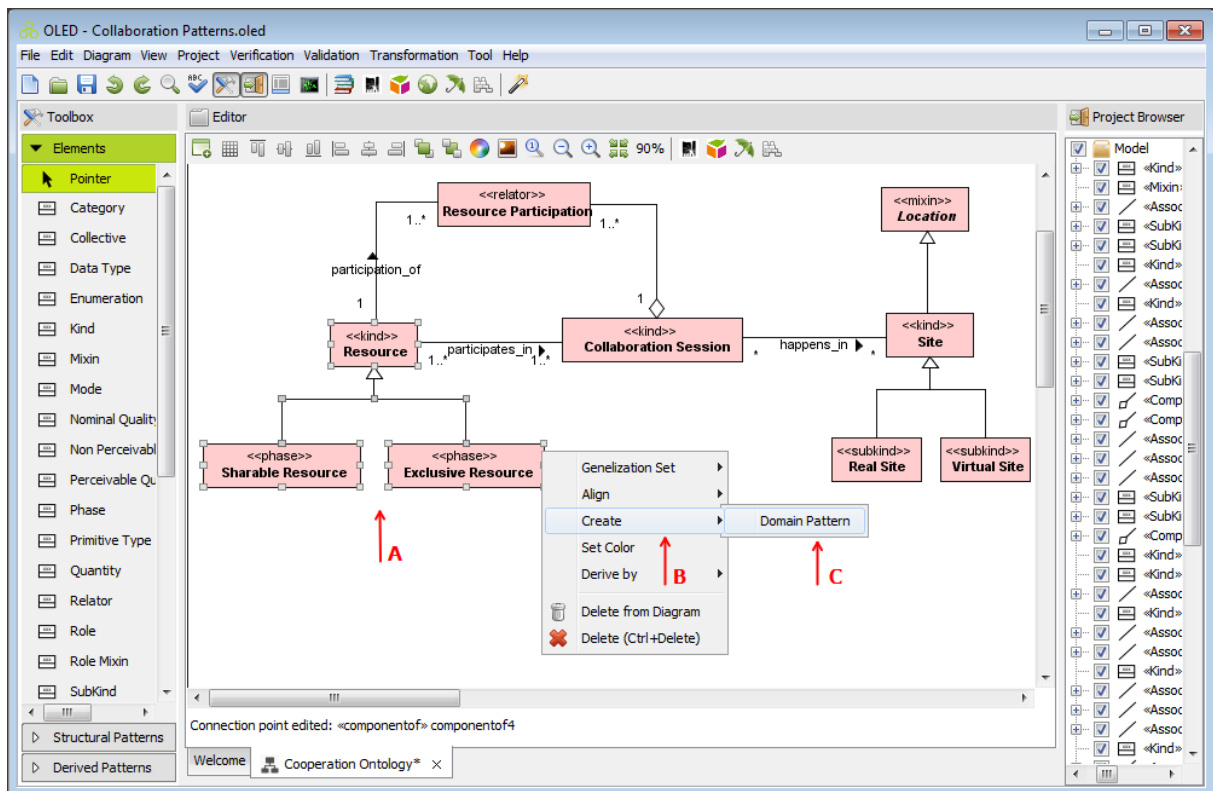


Figura 76 – Criação do DROP (*RN*) a partir de uma ontologia de referência

Caso haja manutenção na ontologia de referência que reflita a necessidade de renomear alguma classe ou troca do estereótipo, por exemplo, o OLED estendido possibilita a ajuste em lote para que a alteração seja aplicada em cada DROP em que a classe está presente, sem a necessidade do engenheiro ter de fazê-la manualmente em todos os lugares. Essa troca também pode feita no próprio DROP e ela será propagada para os outros DROPs em que aparece e na ontologia de referência. Com isso, a manutenção se torna mais suave e o controle das alterações passa ser executado pela ferramenta e não pelo engenheiro, desobrigando o mesmo da responsabilidade pelo rastreo entre as classes e os DROPs. Na Figura 77, essa funcionalidade exemplificada pode ser vista.

O processo e evolução tanto das ontologias de referência quanto dos DROPs do catálogo podem ser realizados com o suporte da ferramenta agilizando e facilitando o

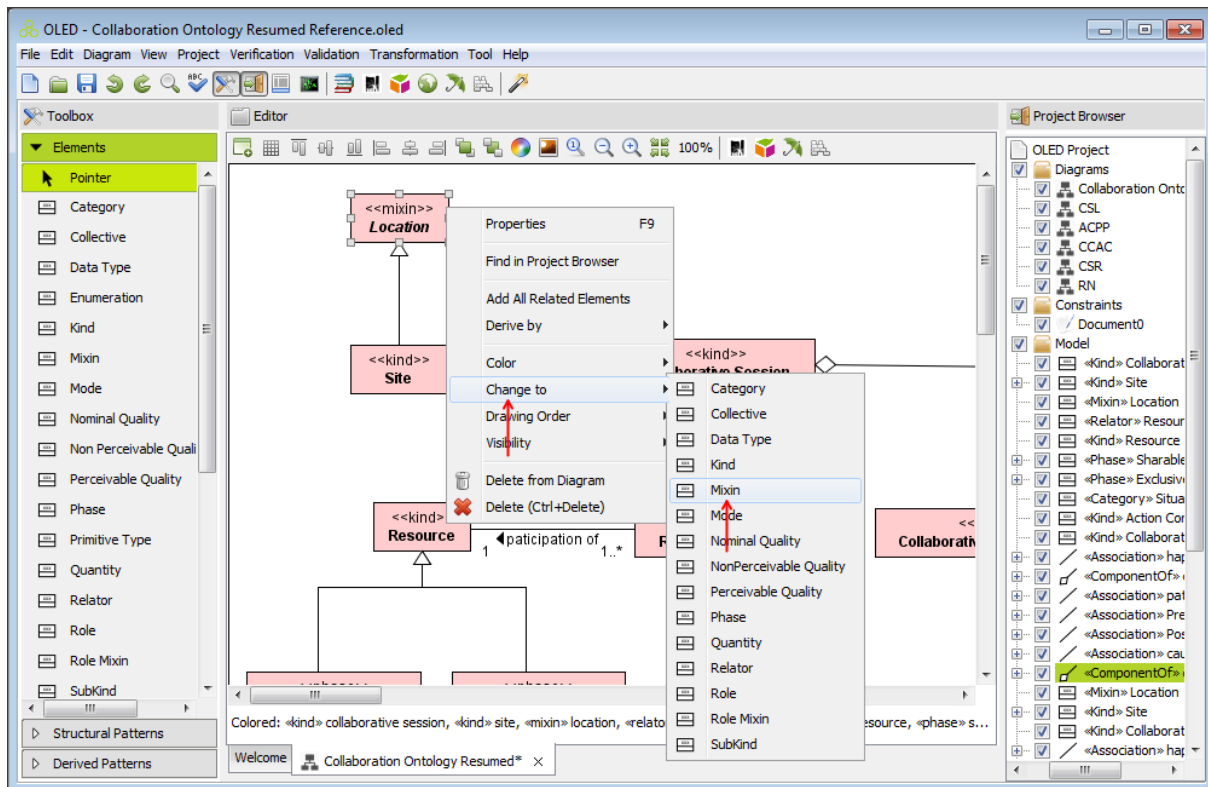


Figura 77 – Alteração em lote de classe no OLED

controle das alterações pelos recursos que ela oferece. O catálogo de DROPs de colaboração se encontra em estágio de construção assim como ajustes na ontologia de referência CONTO. Portanto, o catálogo não possui um processo completo definido para orientar os engenheiros de ontologias. O mesmo pode ser utilizado por meio das questões de competência definidas para os DROPs ou pelo modelo do processo da C-OPL parcialmente implementada (vide Figura 42) conforme apresentado no capítulo 4.

Para exemplificar a utilização do catálogo de DROPs de colaboração foi definida a criação de um editor de textos colaborativos online. Para a construção de uma ontologia operacional para este contexto, serão definidos alguns recursos que esse editor seja capaz de prover. Esses recursos serão materializados por meio de questões, as quais, representam problemas recorrentes que nosso editor deve ser capaz de lidar e conseqüentemente, nossa ontologia operacional do editor deve ser capaz de responder. Essas perguntas estão listadas na Tabela 28.

O primeiro cenário que vamos explorar é a não utilização da C-OPL. Então o mapeamento é realizado comparando as questões levantadas para o editor e as questões de competência definidas para os DROPs. Com o mapeamento feito, passa-se para a etapa de construção da ontologia selecionando os DROPs mapeados para formarem a ontologia operacional do editor de texto colaborativo online. Estes DROPs são os apresentados na seção 3.3. O processo de seleção dos DROPs e adição deles a modelagem pode ser vistos em mais detalhes na seção 5.2. O resultado obtido para a ontologia operacional do editor

Tabela 28 – Mapeamento entre as questões de competência e os DROPs do catálogo de colaboração

Question	Description	Pattern
CQ01	Where does the collaborative editing session happen?	CSL
CQ02	What kinds of collaborative sessions are there?	CSS
CQ03	What kinds of actions are there in a collaborative editing session?	CSA
CQ04	What kinds of commitments are there in the collaborative editing session?	CCAC
CQ05	What are the versions of a document?	ACPPS
CQ06	What changes in document comparing two versions of it?	ACPPS
CQ07	A particular document is available for writing?	RN
CQ08	What documents are sharable?	RN
CQ09	What documents are exclusive?	RN
CQ10	How to write a document?	CSAC
CQ11	Who writes a document?	CCIC

colaborativo online pode ser visto na Figura 78.

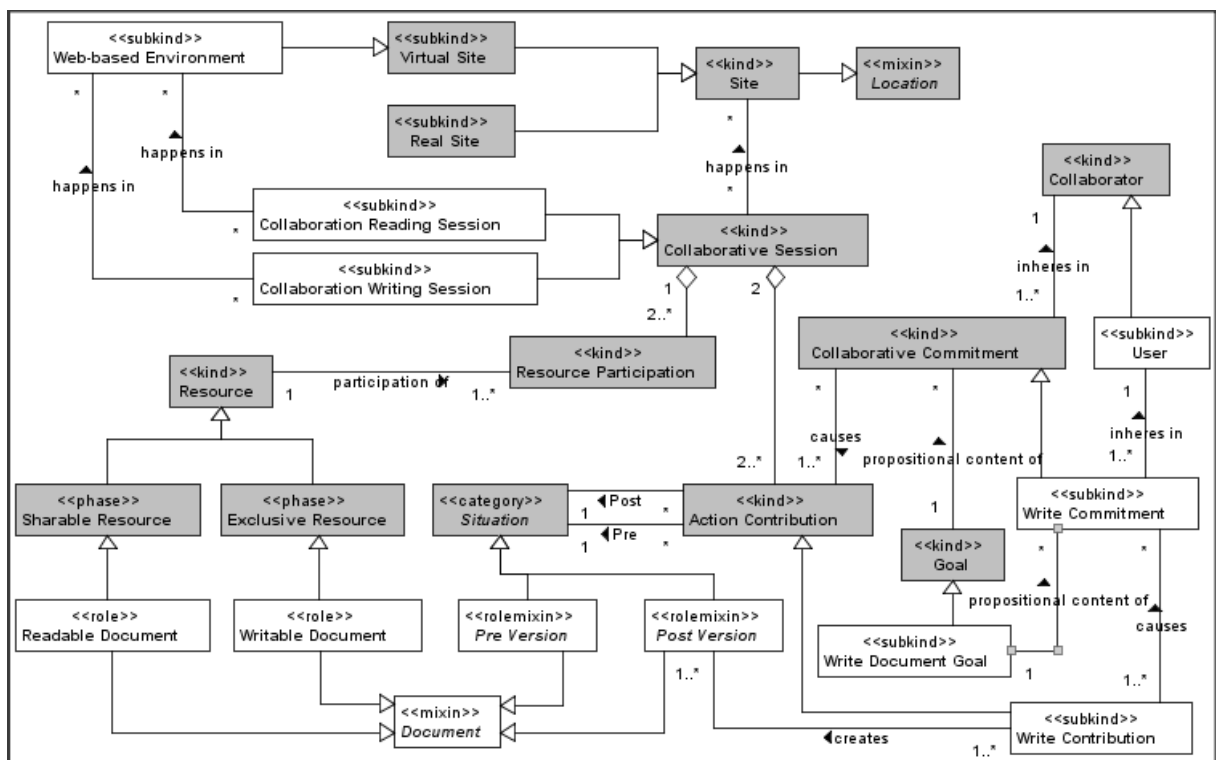


Figura 78 – Ontologia para o editor colaborativo online com a utilização dos DROPs do catálogo de colaboração (SOUZA et al., 2016)

O segundo cenário que vamos explorar é a utilização da C-OPL. Com o modelo do processo da C-OPL, podemos seguir o fluxo de aplicação do DROPs, selecionando aqueles que sejam de interesse. Assim procedendo, será percorrido o caminho no modelo da C-OPL apresentado na Figura 80 destacado em azul. Portanto, basta adicionar a modelagem da ontologia os DROPs e interligá-los para formar a base da ontologia operacional. O

resultado é apresentado na Figura 79. Comparando o resultado alcançado nos dois cenários, basicamente se chegou a mesma modelagem ficando diferente por questões na diferença de versão dos DROPs que estão em constante aprimoramento, mas o ganho de produtividade é perceptível.

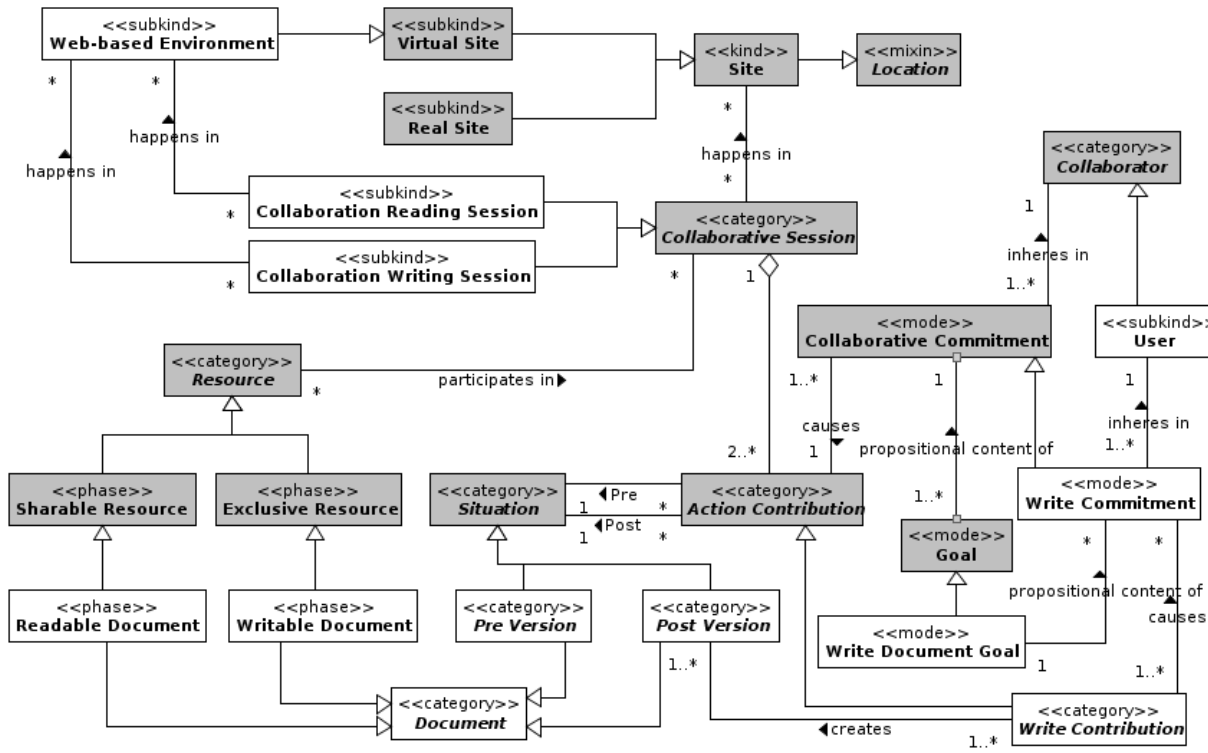


Figura 79 – Ontologia para o editor colaborativo online com a utilização dos DROPs do catálogo de colaboração

Vale reforçar que, para aqueles catálogos que não possuem uma OPL, os engenheiros de ontologias podem fazer a escolha dos DROPs pelas questões de competência, conforme foi feito para o primeiro cenário. Quando o catálogo de DROPs possui uma OPL associada, o modelo do processo dessa OPL pode ser adicionado ao catálogo para servir de guia aos engenheiros de ontologias, assim como foi feito no segundo cenário apresentado. O modelo do processo da OPL ainda não pode ser construído no OLED estendido, devendo ser feito em outra ferramenta e adicionada como uma imagem nas configurações do catálogo.

6.4 Avaliação da Abordagem de Catálogos de DROPs

A avaliação da abordagem de catálogos de DROPs também teve como parte a realização de um experimento em que a mesma foi submetida para utilização na construção de catálogos de DROPs e modelagem de ontologias a partir desses catálogos. Após a utilização, um questionário foi respondido com intuito de encontrar indícios e aprimorar a utilização da abordagem de catálogos de DROPs na modelagem de ontologias. A seguir, é apresentado a pesquisa de avaliação e seus resultados.

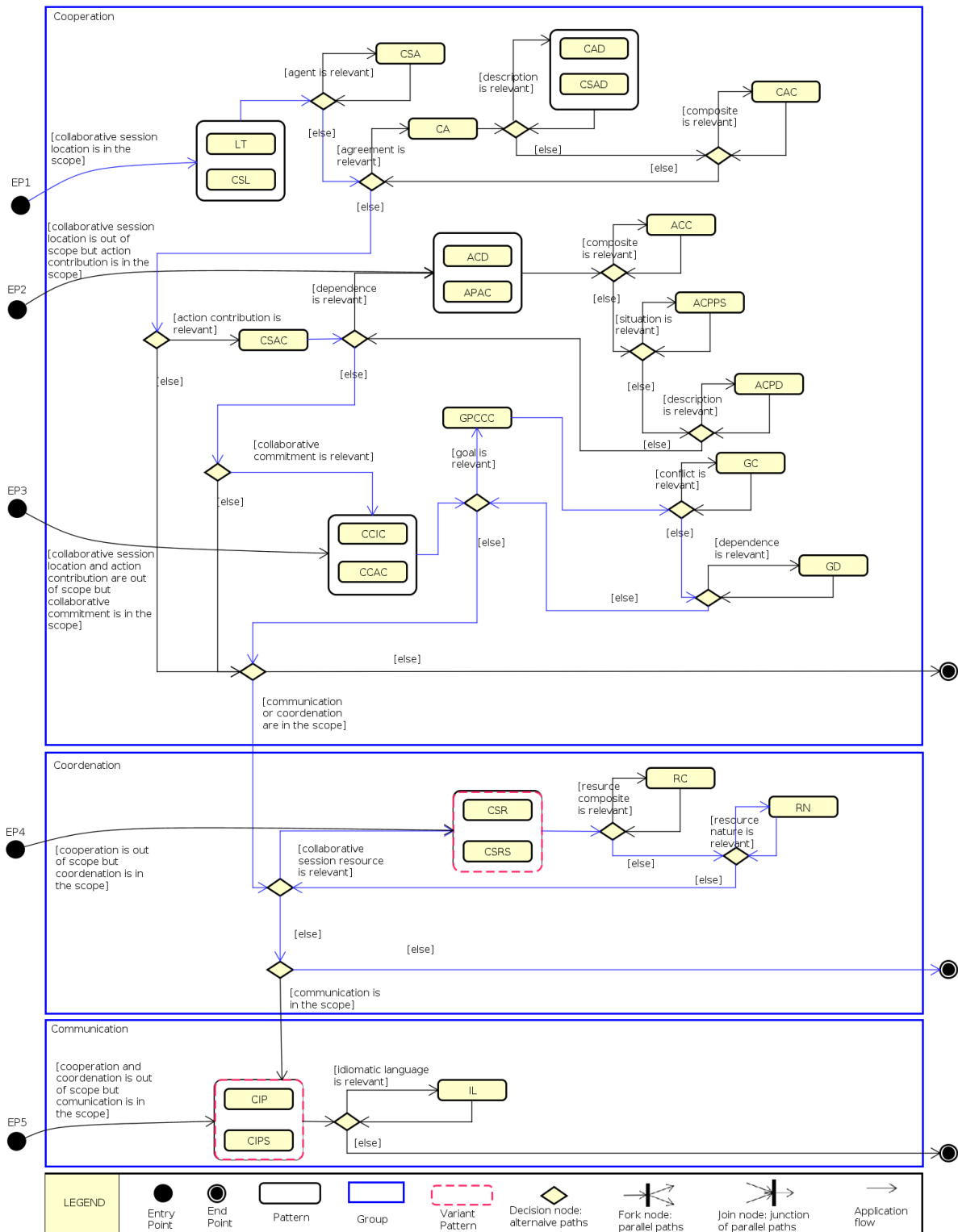


Figura 80 – Modelo do processo da C-OPL para o editor colaborativo

6.4.1 Planejamento da Avaliação

O objetivo da pesquisa realizada foi avaliar a contribuição que a abordagem de catálogos de DROPs com suporte automatizado traz ao processo de modelagem de ontologias operacionais de domínio. Para isso, os participantes tiveram acesso a: (i) um documento contendo os capítulos 4 e 5 para contextualizá-los e guiá-los nas funcionalidades da ferramenta para trabalhar com catálogos de DROPs; (ii) uma versão da ferramenta para que pudessem testar o suporte automatizado criado e a abordagem metodológica para criação de catálogos de DROPs; e (iii) um questionário que permite que os participantes registrem sua percepção após o uso do suporte automatizado para a abordagem de catálogos de DROPs. Esse formulário foi criado com auxílio do Google Forms¹ e é apresentado no Apêndice B.

O questionário inclui questões relacionadas ao tempo de experiência na modelagem de ontologias, o emprego de ferramenta computacional para a modelagem de ontologias, o impacto que a abordagem de catálogos de DROPs traz para a modelagem de ontologias e o suporte automatizado desenvolvido para esse objetivo. O questionário é dotado de questões objetivas em que o participante pode justificar sua resposta expressando sua percepção.

Os participantes da pesquisa foram profissionais com experiência na modelagem de ontologias que poderiam já ter tido contato com catálogos de DROPs ou não bem como alunos e ex-alunos do Programa de Pós-graduação em Informática (PPGI) da UFES com experiência necessária para avaliar a abordagem proposta. Os mesmos foram convidados a participar da avaliação de forma anônima e, não foi estabelecido uma quantidade de mínima ou máxima de participantes, seis foram os que responderam ao questionário de avaliação e, portanto, avaliaram a abordagem de catálogos de DROPs com suporte automatizado. As perguntas do questionário de pesquisa dessa avaliação são apresentadas no Apêndice B. A próxima seção, apresenta os resultados compilados.

6.4.2 Resultados da Avaliação

Nesta seção são apresentados os resultados obtidos para as questões a partir dos questionários respondidos pelos participantes da avaliação da abordagem de catálogos de DROPs. O questionário foi dividido em duas partes em que, a primeira, visa capturar a experiência do participante e, a segunda, visa verificar a percepção do participante quanto a contribuição da abordagem de catálogos de DROPs para a modelagem de ontologias.

¹ <https://www.google.com/forms/about/>

6.4.2.1 Modelagem de Ontologias

A primeira parte do questionário foi idealizada para identificar a experiência do participante, como ele trabalha e o que ele utiliza para modelar ontologias. Os resultados são expressos de forma percentual.

1. Quantos anos de experiência você tem ou teve com modelagem de ontologias?

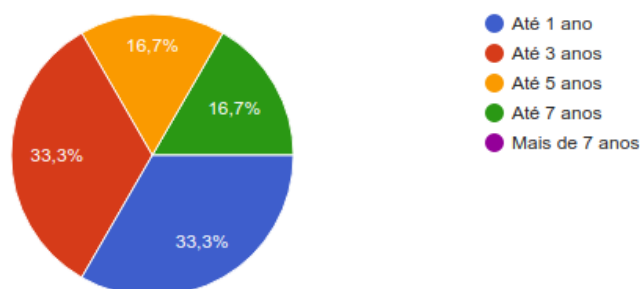


Figura 81 – Tempo de experiência em modelagem de ontologias

A Figura 81 mostra as respostas dadas pelos participantes com relação as experiências que têm em modelagem de ontologias. Dos que participaram da avaliação da abordagem, temos 33% possuindo menos de um ano de experiência; 33% possuindo mais de um e menos de três anos de experiência; 16,7% possuindo mais de três anos e menos de cinco anos; e 16,7% possuindo mais de cinco e menos que sete anos de experiência de modelagem. Assim temos participantes iniciantes, médios e um pouco mais experientes na avaliação. Esses enxergam a modelagem de ontologias atualmente complexa, 100% deles; improdutiva, em 33,3%; vaga, em 16,7%; e produtiva, em 16,7% (vide Figura 82). A complexidade se deve na maior parte das vezes quanto a falta de ferramentas e diretrizes que possam direcionar e auxiliar no processo de modelagem o que impacta também na improdutividade e torna o processo vago, obscuro. Porém, com o aumento no tempo de experiência, a improdutividade e a insegurança, imprecisão que processo dito assim como vago, tendem a diminuir.

2. Como você enxerga a modelagem de ontologias hoje em dia?

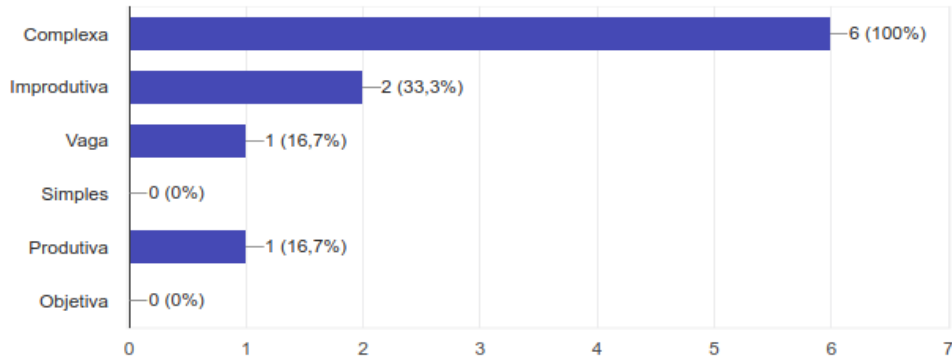


Figura 82 – Percepção da modelagem de ontologias

3. O que você, normalmente, utiliza para a modelagem de ontologias?

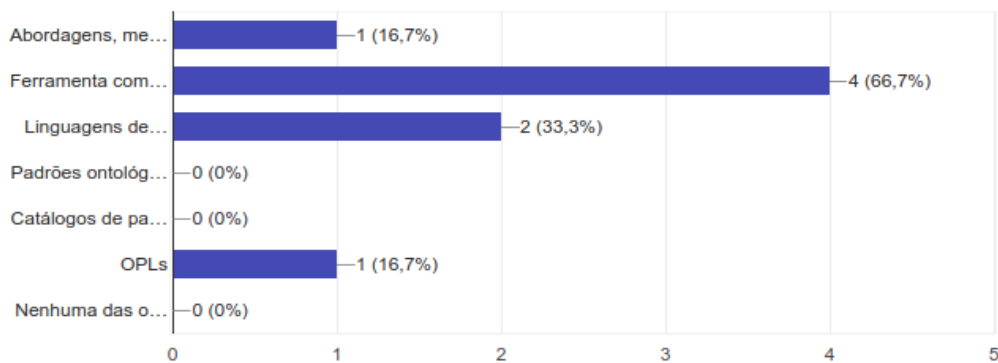


Figura 83 – Recursos utilizados em modelagem de ontologias

Quando os participantes estão modelando ontologias, abordagens, metodologias, etc. são utilizadas por 16,7% deles; ferramentas computacionais, por 66,7%; linguagens de modelagem ontológica como OntoUML, em 33,3%; e OPLs, por 16,7% conforme Figura 83. Padrões ontológicos e catálogos de padrões ontológicos não foram mencionados pelos participantes para a modelagem de ontologias. Quanto a utilização de ferramenta computacional, 66,7% responderam que sempre as utilizam na modelagem de ontologias; 16,7%, utilizam na maioria das vezes e 16,7% raramente (vide Figura 84). As outras opções como "De vez em quando" e "Nunca" não foram escolhidas. Com as informações apresentadas, percebemos a importância do apoio computacional para a modelagem de ontologias e como ela é importante e utilizada no processo de modelagem de ontologias.

4. Com que frequência você utiliza ferramenta computacional para modelagem de ontologias?

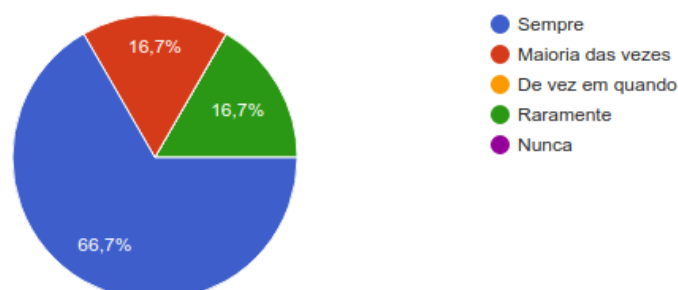


Figura 84 – Suporte utilizado em modelagem de ontologias

5. Quando você utiliza ferramenta computacional para a modelagem de ontologias, essas ferramentas suportam?

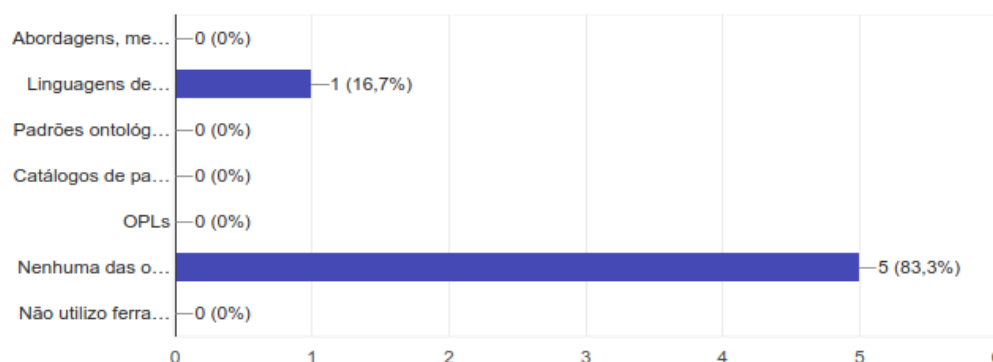


Figura 85 – Recursos presentes na ferramenta computacional para modelagem de ontologias

Quando questionados os participantes quanto a quais recursos essa ferramenta computacional utilizada na modelagem possui, 16,7% responderam que ela suporta linguagens de modelagem ontológicas como OntoUML e que 83,3% responderam que ela não suporta linguagens de modelagem ontológicas como OntoUML (Figura 85), ou seja, possui estruturas que representam elementos de modelagem de ontologias, mas as ferramentas não validam, nem verificam o que está sendo modelado, deixando a responsabilidade pela validade da ontologia modelada completamente a cargo do modelador. Para confirmar a percepção dos participantes quanto a utilização de ferramenta computacional na modelagem de ontologias, 100% deles intendem que ferramenta computacional é de muita utilizada e importância no processo de modelagem de ontologias conforme apresentado na Figura 86.

6. Qual a sua avaliação da utilização de ferramenta computacional para apoiar a modelagem de ontologias?

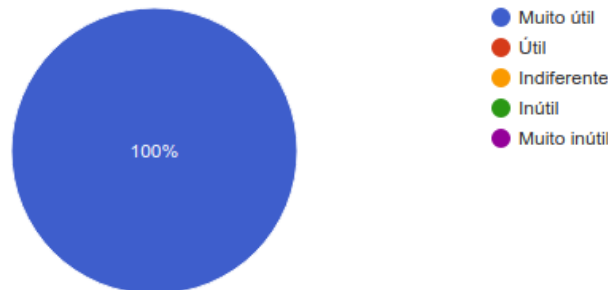


Figura 86 – Percepção do suporte computacional em modelagem de ontologias

6.4.2.2 Abordagem de Catálogo de DROPs

A segunda parte do questionário foi idealizada para identificar a percepção dos participantes quanto a contribuição adicionada a modelagem de ontologias feita a partir de catálogos de DROPs com suporte automatizado. Os resultados são expressos de forma percentual.

7. Qual a sua avaliação do suporte automatizado implementado no OLED e da abordagem metodológica proposta para criação de catálogos de DROPs?

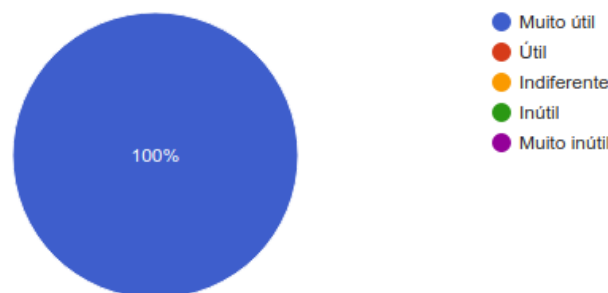


Figura 87 – Avaliação do suporte automatizado e da abordagem metodológica para catálogo de DROPs

Quando questionados sobre a avaliação que os participantes faziam da abordagem metodológica (capítulo 4) e do suporte automatizado (capítulo 5) quanto ao processo criação de catálogos de DROPs, 100% dos participantes avaliaram que os mesmos são muito úteis. As demais opções não foram mencionadas conforme Figura 87. As justificativas apresentadas, em resumo, foram: proporcionar padronização, agilidade e praticidade ao

processo de criação de catálogos de DROPs; utilizar ferramentas apropriadas facilitam bastante o trabalho e também garantem a qualidade da ontologia uma vez que as mesmas têm verificações para isso; contribuir com a produtividade e com a manutenção da modelagem uma vez que pode ser alterado somente o que for de interesse, não necessitando refazer toda a modelagem caso não fosse utilizado ferramenta computacional; e fornecer funcionalidades que auxiliam na modelagem de ontologias.

8. Em sua avaliação, o suporte automatizado implementado no OLED para criação de catálogos de DROPs e a abordagem metodológica proposta favorecem a produtividade na criação desses catálogos?

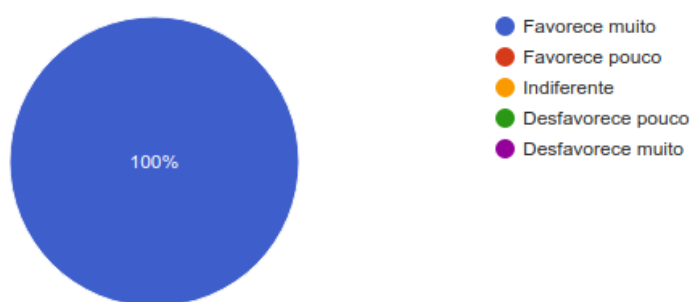


Figura 88 – Avaliação do suporte automatizado e da abordagem metodológica quanto a produtividade para catálogo de DROPs

Os participantes, 100% deles (vide Figura 88), avaliaram que tanto a abordagem metodológica (capítulo 4) quanto o suporte automatizado (capítulo 5) favorecem muito a produtividade para a criação de catálogos de DROPs. As justificativas apresentadas, em resumo, foram: disponibilizar funcionalidade para extração de DROPs a partir de ontologias de referência e para configuração de DROPs; a abordagem fornece um passo a passo para a criação de padrões e o suporte fornece funcionalidades para trabalhar com catálogos de DROPs; fornece uma forma sistemática e automatizada de extração de padrões a partir de uma ontologia; e favorece a produtividade, pois a abordagem fornece os passos que devem ser seguidos para se criar um catálogo e a ferramenta também ajuda muito na produtividade, uma vez que a partir de uma ontologia de referência é possível selecionar e criar os padrões e com base nesses padrões criar o catálogo. Com certeza é mais rápido montar um catálogo na ferramenta do que em um documento de texto/tabelas, por exemplo.

Os participantes, 100% deles (vide Figura 89), avaliaram que tanto a abordagem metodológica (capítulo 4) quanto o suporte automatizado (capítulo 5) favorecem muito a qualidade dos catálogos de DROPs. As justificativas apresentadas, em resumo, foram: devido a sistematização do processo de definição dos DROPs; devido aos passos da abordagem, os padrões são validados/testados antes de serem incorporados no catálogo;

9. Em sua avaliação, o suporte automatizado implementado no OLED para criação de catálogos de DROPs e a abordagem metodológica proposta favorecem a qualidade desses catálogos e dos DROPs deles?



Figura 89 – Avaliação do suporte automatizado e da abordagem metodológica quanto a qualidade dos catálogos de DROPs

fornece uma forma sistemática e automatizada de extração de padrões a partir de uma ontologia e a validação do padrões; e porque a abordagem metodológica padroniza a forma de criação dos padrões do catálogo e a ferramenta sistematiza a estrutura dos catálogos e as configurações dos mesmos.

10. Em sua avaliação, o suporte automatizado implementado no OLED para criação de catálogos de DROPs e a abordagem metodológica proposta favorecem o compartilhamento desses catálogos e dos DROPs deles?



Figura 90 – Avaliação do suporte automatizado e da abordagem metodológica quanto ao compartilhamento dos catálogos de DROPs

Também 100% dos participantes (vide Figura 90) avaliaram que tanto a abordagem metodológica (capítulo 4) quanto o suporte automatizado (capítulo 5) favorecem muito o compartilhamento dos catálogos de DROPs. As justificativas apresentadas, em resumo, foram: devido aos catálogo de padrões estarem empacotado, pronto para serem reutilizados;

porque os passos para a criação de padrões padronizam a forma de criação deles e a estruturação dos mesmos em catálogos fornece uma estrutura de organização que, apoiada na ferramenta, possibilita a reutilização dos catálogos em modelagens; devido aos catálogos de DROPs serem criados utilizando uma abordagem metodológica, a definição dos DROPs obedece uma padronização. Logo o compartilhamento dos mesmos se torna mais prático. Aliado a isso, o empacotamento do catálogo de DROPs pela ferramenta, faz com que esse catálogo seja compartilhado e utilizado em várias modelagens.

11. Em sua avaliação, o suporte automatizado implementado no OLED para criação de catálogos de DROPs e a abordagem metodológica proposta favorecem a configuração desses catálogos e dos DROPs deles?



Figura 91 – Avaliação do suporte automatizado e da abordagem metodológica quanto a configuração dos catálogos de DROPs

100% dos participantes (vide Figura 91) também avaliaram que tanto a abordagem metodológica (capítulo 4) quanto o suporte automatizado (capítulo 5) favorecem muito a configuração dos catálogos de DROPs. As justificativas apresentadas, em resumo, foram: devido às funcionalidades e à forma de organização das propriedades para configuração dos padrões do catálogo; devido ser possível, uma vez que em uma única tela, preencher todas as informações dos padrões necessárias, desde informações básicas como nome do padrão até questões de competências; todas as configurações de fácil acesso em uma mesma tela e de forma bem organizada. A abordagem também deixa bem estruturada também a criação de DROPs a partir de ontologias de referência.

Quando questionados quanto ao suporte automatizado criado para a reutilização de catálogos de DROPs a fim de modelagem de ontologias com os mesmos, 100% (Figura 92) dos participantes avaliaram que ele é muito útil. As demais opções não foram mencionadas. As principais justificativas apresentadas, em resumo, foram: permitir que um catálogo seja criado uma única vez e este seja utilizado por várias pessoas, por exemplo, uma equipe que esteja construindo uma ontologia; possibilitar que a modelagem não seja feita do nada ou do início podendo ser utilizados os padrões do catálogo inserindo um novo nível

12. Qual a sua avaliação do suporte automatizado implementado no OLED para reutilização de catálogos de DROPs?

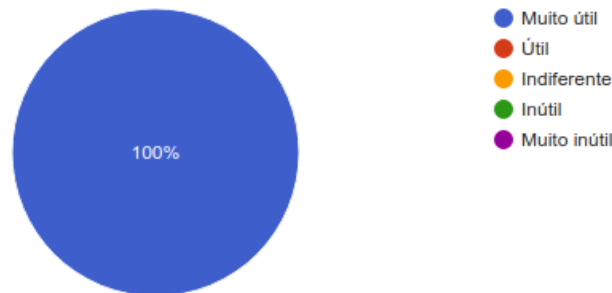


Figura 92 – Avaliação do suporte automatizado na reutilização de catálogos de DROPs

de abstração na modelagem; porque o modelador pode utilizar os padrões para criar a ontologia ao invés de precisar fazer isso do nada, ganhando produtividade; e porque os catálogos de DROPs importados e disponibilizados na caixa de ferramentas ficam a mão de forma prática mostrando um resumo das principais propriedades dos DROPs, auxiliando a escolha dos mesmos para a utilização.

13. Em sua avaliação, o suporte automatizado implementado no OLED para reutilização de catálogos de DROPs favorece produtividade na modelagem de ontologias?



Figura 93 – Avaliação do suporte automatizado para reutilização de catálogo de DROPs na modelagem de ontologias quanto à produtividade

Os participantes também foram questionados como o suporte automatizado implementado no OLED para a reutilização de catálogos de DROPs favorece, em alguns aspectos, a modelagem de ontologias. O primeiro deles, diz respeito a produtividade. 100% dos participantes (vide Figura 93) responderam que favorece muito a produtividade uma vez que, para eles: a modelagem de ontologias a partir de padrões, aumentando o nível de abstração e diminuindo o trabalho na modelagem devido a ferramenta "desenhar" os

padrões escolhidos e não modelador; porque tendo um catálogo de ontologias, a modelagem de uma ontologia de domínio é mais rápida, uma vez que o modelador de ontologia irá reusar soluções prontas; porque os catálogos possuem soluções de modelagem testadas e estruturadas para domínios que o modelador pode não ser especialista, ajudando-o na atividade; e por entregar aos engenheiros de ontologias soluções de problemas recorrentes de modelagem diminuindo a necessidade de um especialista no domínio.

14. Em sua avaliação, o suporte automatizado implementado no OLED para reutilização de catálogos de DROPs favorece a qualidade das ontologias criadas com eles?

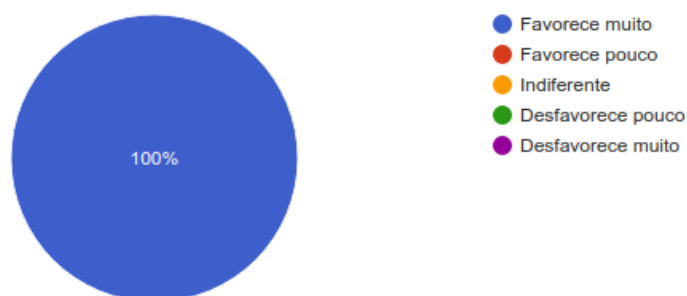


Figura 94 – Avaliação do suporte automatizado para reutilização de catálogo de DROPs na modelagem de ontologias quanto à qualidade

Quando questionados quanto a qualidade das ontologias criadas a partir dos catálogos de DROPs, os participantes, 100% deles (vide Figura 94) avaliaram que favorece muito a qualidade uma vez que, em resumo, para eles: os padrões contidos nos catálogos foram testados e validados por especialistas, assim as ontologias criada a partir deles, carregam as características dos padrões; porque as ontologias criadas a partir dos catálogos herdaram as características dos padrões utilizados, conseqüentemente, se eles forem bem feitos, as ontologias também serão e sendo verdadeiro também o contrário; e porque a ferramenta sistematiza a modelagem "desenhando" os padrões ligando uns aos outros para formar a base da ontologia, diminuindo a possibilidade de erro e inconsistências na modelagem.

100% dos participantes também avaliaram que o suporte automatizado para reutilização de catálogos de DROPs favorece o compartilhamento das ontologias conforme Figura 95. As justificativas apresentadas, em resumo, foram: que uma ontologia deve fornecer um conceituação compartilhada e que catálogos de DROPs formados por padrões que carreguem conceituações compartilhadas, levam a ontologias que obedecem o mesmo critério; como o catálogo de DROPs pode ser reutilizado, as ontologias construídas com ele, são compartilhadas porque o conhecimento do domínio está nos padrões do catálogo; desde que os catálogos de padrões sejam criados com o conhecimento compartilhado do domínio

15. Em sua avaliação, o suporte automatizado implementado no OLED para reutilização de catálogos de DROPs favorece o compartilhamento de ontologias construídas com esses catálogos?



Figura 95 – Avaliação do suporte automatizado para reutilização de catálogo de DROPs na modelagem de ontologias quanto ao compartilhamento

de interesse, as ontologias modeladas com ele também serão; e as ontologias criadas com os DROPs de catálogos serão mais familiares a uma comunidade uma vez que os DROPs do catálogo serão padrões conhecidos dentro dessa comunidade.

16. Em sua avaliação, a abordagem metodológica proposta para a criação de catálogos de DROPs e o suporte automatizado implementados no OLED adicionam ao processo de construção de catálogo de DROPs mais?

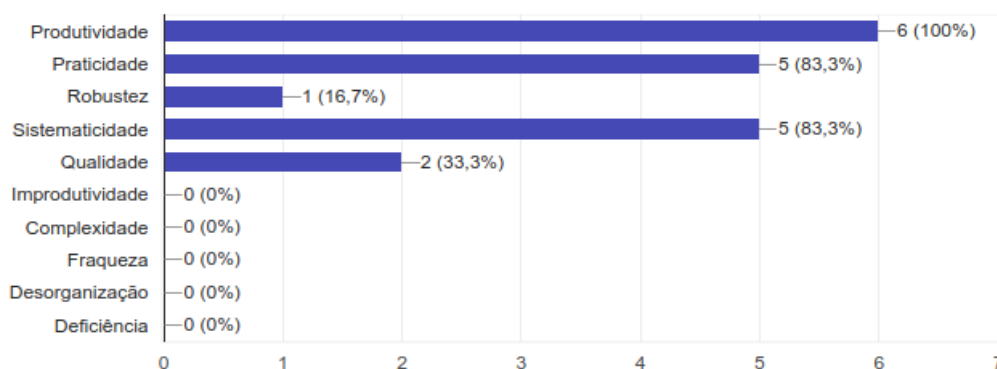


Figura 96 – Avaliação do suporte automatizado e da abordagem metodológica para catálogos de DROPs

No geral, quando questionados sobre a contribuição que a abordagem metodológica (capítulo 4) e o suporte automatizado (capítulo 5) dão para a construção de catálogos de DROPs, para 100% dos participantes, adiciona produtividade; para 83,3%, praticidade e sistematicidade; para 33,3%, qualidade; e para 16,7% robustez (Figura 96).

17. Em sua avaliação, a abordagem metodológica proposta para catálogos de DROPs e o suporte automatizado implementado no OLED adicionam ao processo de modelagem de ontologias mais?

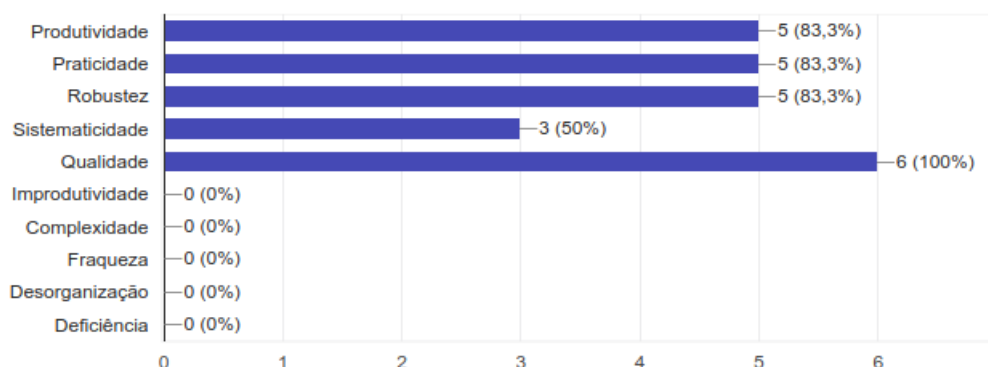


Figura 97 – Contribuição da abordagem metodológica e do suporte automatizado para modelagem de ontologias

Quando questionados sobre a contribuição adicionada ao processo de modelagem de ontologias pela abordagem metodológica (capítulo 4) e pelo suporte automatizado (capítulo 5), 83,3% dos participantes perceberam que o processo de modelagem de ontologias se torna mais produtivo, prático e robusto; 50% que o processo se torna mais sistemático; e 100% deles que o processo de modelagem de ontologias ganha mais qualidade. As demais opções não foram percebidas pelos participantes conforme pode ser visto na Figura 97. Isso é interessante porque nenhum dos participantes haviam utilizado a abordagem de catálogos de DROPs com suporte automatizado para modelagem de ontologias, ou seja, 100% dos participantes haviam experimentado pela primeira vez essa forma de modelagem de ontologias (vide Figura 98).

A contribuição da abordagem metodológica e do suporte automatizado apresentados nos capítulos 4 e 5 respectivamente fariam com que 100% dos participantes as utilizassem para a construção de catálogos de DROPs, conforme Figura 99. Os motivos apresentados, em resumo, foram: tornar o processo mais produtivo, prático e sistemático; permitir reaproveitar estruturas testadas e validadas; fornecer uma abordagem sistemática para a criação de catálogos de DROPs que por sua vez adiciona um nível maior de abstração na modelagem de ontologias aumentando a produtividade por meio da sistematização do processo; apresenta uma abordagem que padroniza a forma de definir padrões e, o suporte, torna o processo de modelagem mais eficiente, favorecendo a manutenção da ontologia.

No quesito de contribuição do suporte automatizado e a utilização da abordagem de catálogo de DROPs para a modelagem de ontologias, 100% dos participantes intendem que há contribuição conforme Figura 100. Alguns dos motivos apresentados, em

18. Você já havia utilizado a abordagem de catálogos de DROPs com suporte automatizado para modelagem de ontologias?

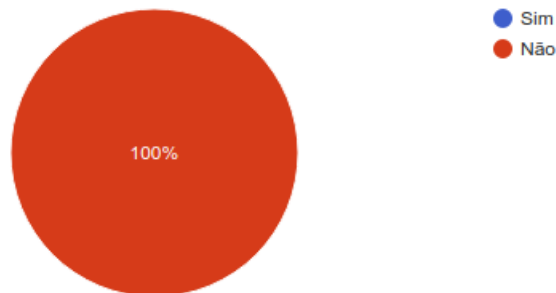


Figura 98 – Familiaridade com a abordagem de catálogos de DROPs com suporte automatizado

19. Em sua avaliação, no geral, você utilizaria a abordagem metodológica para a criação de catálogos de DROPs e o suporte automatizado implementado para a criação de catálogos e para a modelagem de ontologias?

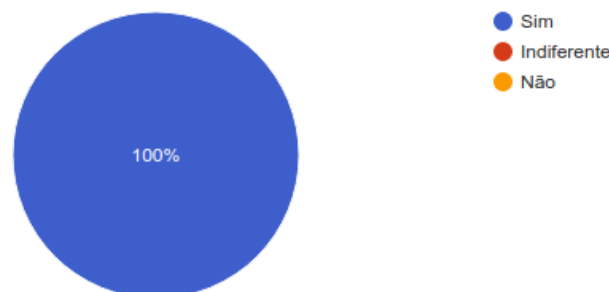


Figura 99 – Contribuição da abordagem metodológica e do suporte automatizado para catálogos de DROPs

resumo, são: possibilitar a construção de ontologias a partir de padrões preestabelecido e reutilização dos mesmos; permitir reaproveitar de forma prática estruturas já definidas em outras modelagens e evitando modelagem incorreta; fornecer uma abordagem sistemática para a criação e reutilização de catálogos de DROPs que adicionam um nível maior de abstração na modelagem, aumentando a produtividade, qualidade e eficiência do processo; permitir criar ontologias a partir de estruturas bem definidas, os padrões, o que ajuda na qualidade das ontologias aliado ao suporte que proporciona mais produtividade.

20. Em sua avaliação, no geral, o suporte automatizado e a abordagem metodológica de catálogos de DROPs contribuem com o processo de modelagem de ontologias?

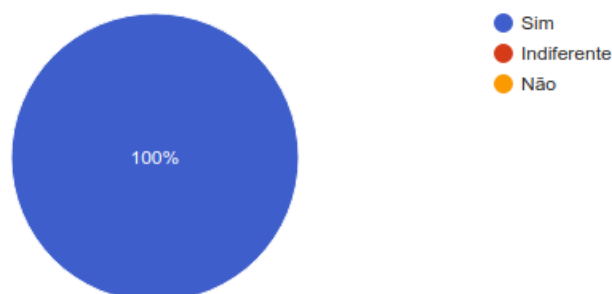


Figura 100 – Contribuição da abordagem de catálogos de DROPs e o suporte automatizado para modelagem de ontologias

6.5 Considerações

Os exemplos de utilização apresentados neste capítulo visaram mostrar como a abordagem de catálogo de DROPs pode ajudar na modelagem de ontologias. Para os três domínios escolhidos, serviço (seção 6.1), processo de software baseado na ISO (6.2) e colaboração (seção 6.3), comparou-se ontologias modeladas nestes domínios com as mesmas ontologias sendo criadas com a abordagem de catálogos de DROPs e o suporte automatizado propostos neste trabalho. Os resultados foram os mesmos, a menos de pequenas diferenças de versões de especificações dos padrões utilizados. Foi visto também que as OPLs podem se beneficiar bastantes com o suporte automatizado e com a própria abordagem de catálogos de DROPs uma vez que as OPLs são conjuntos de DROPs ligados (catálogo de DROPs) de forma direcionada, traduzido no modelo do processo da OPL.

Quanto a pesquisa, em alguns aspectos ela foi um tanto repetitiva, como as questões de 8 a 11 que poderiam ter sido avaliadas em uma única com as opções organizadas de forma semelhante a questão 16, mas isso fez com que se conseguisse um nível maior de detalhe da avaliação dos participantes quanto a abordagem metodológica e do suporte automatizado apresentados nos capítulos 4 e 5. Assim, foi possível identificar, quanto modelagem de ontologias, que os participantes possuíam graus de experiência diferentes, a maioria utilizava ferramenta computacional, mas essa não possuía suporte a linguagens ontológicas com as quais seria possível realizar validações, pela ferramenta, da modelagem.

Ainda para os participantes, o suporte automatizado provido (capítulo 5) e a abordagem metodológica proposta (capítulo 4) para criação de catálogos de DROPs favorecem muito a produtividade, a qualidade, o compartilhamento e a configuração dos catálogos pelos motivos apresentados na seção anterior. Da mesma forma, aconteceu

com a reutilização de catálogos de DROPs para a modelagem de ontologias, em que a avaliação da abordagem e do suporte favorecem muito a produtividade, a qualidade e o compartilhamento das ontologias criadas a partir dos catálogos.

Por fim, os participantes responderam que estavam tendo contato, pela primeira vez, com a modelagem de ontologias a partir de catálogos de DROPs com suporte automatizado. Como resultado da avaliação, a abordagem de catálogos de DROPs foi considerada agregadora de valor para a construção de catálogos de DROPs e para a modelagem de ontologias quando utilizados catálogos de DROPs. Esses participantes responderam que utilizariam a abordagem para modelar ontologias. Para maior parte deles, é importante que tenham um catálogo disponível com os DROPs para facilitar para o engenheiro de ontologias. Essa abordagem e observação anterior abrem espaço para a atuação dos engenheiros de catálogos de DROPs contribuindo com catálogos estruturadas e bem definidos.

7 Trabalhos Relacionados

Alguns trabalhos têm sido realizados no campo de reutilização de ontologias apoiado por ferramentas computacionais. Cada um deles tem suas respectivas metodologias e escopo de atuação. A seguir serão apresetados alguns deles e pontuadas algumas diferenças que tornam a abordagem de catálogo de DROPs implementada no OLED estendido uma ferramenta poderosa de auxílio aos engenheiros de ontologias.

7.1 NeOn Toolkit

O *NeOn Toolkit* é um ambiente de engenharia de ontologia para suporte ao ciclo de vida da Engenharia de Ontologias. Ele é projetado em torno de uma arquitetura aberta e modular, que inclui serviços de infraestrutura, como um registro e um repositório, e suporta componentes distribuídos para gerenciamento de ontologia, raciocínio e colaboração em ambientes em rede. Uma forte ênfase é colocada no gerenciamento de ontologia em rede, ou seja, suporte para Engenharia de Ontologias incorporadas em uma rede de ontologias através de relações semânticas ricas, incluindo modelos de ontologias modulares e mapeamentos em ontologias (HAASE et al., 2008; D'AQUIN et al., 2009).

Apesar de o OLED estendido ter semelhanças com o *NeOn Toolkit*, ambas detêm algumas diferenças importantes. A primeira delas se refere a linguagem em que a modelagem é realizada. O OLED estendido utiliza OntoUML, que é um perfil ontologicamente bem fundamentado do diagrama de classes UML 2.0 com o acréscimo de uma definição semântica formal precisa por se tornar isomórfico à UFO. Já o *NeOn Toolkit* utiliza OWL, é uma linguagem da Web Semântica projetada para representar conhecimento rico e complexo sobre coisas, grupos de coisas e relações entre coisas. Na prática, construir ontologias com o OLED estendido se assemelha a construir modelos UML, enquanto construir ontologias no com o *NeOn Toolkit*, se assemelha a fazer uma descrição hierárquica que pode incluir classes e suas respectivas propriedades e seus relacionamentos.

A segunda diferença está relacionada à reutilização de ontologias. O OLED estendido foi concebido para reutilizar ontologias na forma de OPs que é a menor porção de uma ontologia que pode ser reutilizada. Os OPs são responsáveis por capturar o conhecimento sobre determinado problema de modelagem para posterior utilização. O *NeOn Toolkit* trabalha a reutilização em ontologias por meio do reaproveitamento de classes presentes em outras ontologias cadastradas em repositórios. Essas classes podem ser reutilizadas por meio de consulta em repositórios de ontologias. De posse do resultado, uma lista com aqueles possíveis candidatados e mostrada para a seleção do desejado.

Um terceiro ponto de diferença se refere ao propósito a que cada um foi construído. O OLED estendido tem por propósito trabalhar dentro de domínios específicos, fornecendo ferramentas para a criação de catálogos de DROPs e utilização destes em modelagem de domínio. Ele tem um escopo mais restrito quanto ao universo de domínios de modelagem. O NeOn Toolkit possui um propósito mais geral para a construção de ontologias sem um foco em domínios específicos. A diferença principal é que os padrões com os quais o NeOn Toolkit trabalha são de implementação, ao invés de serem de DROPs, já que são baseados em OWL. A construção de ontologias se dá pela criação, registro e resgate de ontologias que estão agrupadas em repositórios. Um trabalho relacionado ao NeOn Toolkit pode ser visto em ODP, onde são encontradas algumas ontologias¹.

Por fim, outra diferença se apresenta na forma de seleção das classes para a composição da ontologia. No NeOn Toolkit, as pesquisas nos repositórios de ontologias são feitas por meio de um *plugin*, Watson, e o engenheiro de ontologias deverá avaliar o resultado e escolher, de acordo com o entendimento dele, o que aproveitar, qual classe adiciona na modelagem. No OLED estendido, os catálogos são construídos e disponibilizados com a própria ferramenta e neles também podem ser adicionados, no momento da implementação, um processo de seleção dos DROPs pertencentes ao catálogo, tornando um processo guiado, e facilitando o processo de escolha. Outro ponto é que o OLED estendido adiciona a ontologia em processo de modelagem todas as classes pertencentes ao DROPs escolhido, poupando tempo e esforço.

7.2 WebODE

O WebODE é um ambiente de trabalho escalável e integrado para Engenharia de Ontologias que visa facilitar a representação, o raciocínio e a troca de ontologias com outras ferramentas de ontologia e aplicações baseadas em ontologia. Foi desenvolvido pelo Grupo de Ontologia da Universidade Técnica de Madri. O modelo de conhecimento da WebODE (ARPÍREZ et al., 2001) baseia-se nas representações intermediárias propostas no *Methontology framework* (LÓPEZ et al., 1999). Assim, permite conceitos de modelagem e seus atributos (atributos de classe e de instância), taxonomias de conceitos, partições de classes disjuntas e exaustivas, relações binárias entre conceitos, propriedades de relações, constantes, axiomas e instâncias de conceitos e relações.

O OLED, assim como o WebODE, são ferramentas para apoiar o desenvolvimento de ontologias, mas cada uma realiza esta tarefa de forma diferentes. Por exemplo, é possível perceber que o WebODE baseia-se em dois pilares: o *Methontology framework* que fornece uma abordagem para aquisição, construção e validação de modelos de conhecimento de domínio por engenheiros de ontologias não especialistas no domínio. O outro é o ODE

¹ <[<http://ontologydesignpatterns.org/wiki/Ontology_Design_Patterns_._org_\(ODP\)>](http://ontologydesignpatterns.org/wiki/Ontology_Design_Patterns_._org_(ODP))>

que suporta esta estrutura, permitindo que ontologias sejam construídas no nível de conhecimento e implementadas automaticamente usando tradutores. Também existem alguns *plugins* para utilização junto ao WebODE como o ODEClean, que permite que os desenvolvedores avaliem as taxonomias usando método OntoClean (GUARINO; WELTY, 2002; FERNÁNDEZ-LÓPEZ; GÓMEZ-PÉREZ; GUARINO, 2001; WELTY; ANDERSEN, 2005). Como já mencionado, o OLED utiliza OntoUML, que é baseada em UFO e, assim, utiliza-se do potencial desta ontologia de fundamentação para criar ontologias de qualidade.

No que tange a reutilização de ontologias, o OLED estendido reutiliza ontologias por meio de DROPs que são extraídos de ontologias de referências de domínio e que capturam o conhecimento de um problema recorrente de modelagem nesse domínio para posterior utilização. Com o WebODE, a reutilização em ontologias se dá por meio reaproveitamento de outras ontologias cadastradas em repositórios e disponibilizadas por meios de serviços que são acessadas pela ferramenta pelo engenheiro de ontologias. Desta forma, fica a cargo do engenheiro de ontologias a busca, avaliação e escolha do que reutilizar. Como estrutura de organização o OLED estendido utiliza o catálogo para o agrupamento do DROPs enquanto o WebODE utiliza o repositório para agrupar as ontologias.

Finalmente, pelo fato do OLED estendido trabalhar com catálogos de DROPs, a modelagem de ontologias de domínio nos quais existem catálogos implementados, se torna um processo enxuto devido à possibilidade de um processo de seleção dos DROPs ser anexado ao catálogo para servir de guia ao engenheiro de ontologias, identificando inclusive quais DROPs utilizar dependendo do propósito desejado para a ontologia em modelagem. Com o WebODE, até onde foi possível perceber, a construção de ontologias se dá por um processo para aquisição, construção e validação de modelos de conhecimento de domínio e, posteriormente, a geração de ontologia baseada neste processo. A criação de ontologias parte sempre da aquisição do conhecimento.

7.3 Protégé

O Protégé (ERIKSSON et al., 1995; MUSEN et al., 1995) é um ambiente para desenvolvimento de sistemas baseados em conhecimento e possui uma crescente comunidade de usuários que contribuem adicionando novas capacidades e orientando a evolução da ferramenta. O Protégé tem, entre suas funcionalidades, a capacidade de construir modelos de domínio e aplicativos baseados em conhecimento com ontologias (MOLINA et al., 1996). Em sua versão Web, WebProtégé, criou-se um ambiente flexível para a criação, *upload*, modificação e o compartilhamento de ontologias para visualização e edição colaborativa. Os recursos de colaboração incluem compartilhamento e permissões, notas e discussões segmentadas, relógios, notificações por e-mail assim como RDF/XML, Turtle, OWL/XML, OBO e outros formatos disponíveis para o *upload* e *download* de ontologias (PROTÉGÉ,

2017).

Algumas diferenças importantes podem ser notadas entre o OLED estendido e o Protégé. A abordagem Protégé-II (MUSEN et al., 1995), que introduz ontologias no Protégé, baseia-se em três passos principais na construção do modelo de conhecimento: (i) a formulação do método ontológico que inclui o conjunto de funções de conhecimento utilizadas pelo método, (ii) a definição da ontologia de domínio (de alguma forma independente de método) onde o método será aplicado, e (iii) a formulação de relações de mapeamento relacionando ambas as ontologias (MOLINA et al., 1996). Na abordagem utilizada no OLED estendido, a própria modelagem utilizando OntoUML detém toda a base de conhecimento sobre o domínio. O mapeamento que existe entre UML e OntoUML fica implícito e carrega os benefícios do uso de UFO.

Com relação à reutilização de ontologias, o OLED estendido implementa a abordagem de catálogo de DROPs como forma de criar um ambiente propício para a modelagem de ontologias a partir de OP organizados e estruturados na forma de catálogos. O Protégé-II implementa uma abordagem de reutilização, até onde foi possível avaliar, por meio da base de conhecimento do domínio e também por meio de busca de seleção de ontologias em repositórios. Isso leva o engenheiro de ontologias a ter um conhecimento maior do domínio em que a modelagem se desenvolve para poder avaliar o conhecimento e as ontologias presentes no repositório para reutilização. Fica a cargo do engenheiro de ontologias avaliar quais partes da base de conhecimento e ontologias é de interesse para a modelagem.

7.4 Considerações Relevantes

Todos esses trabalhos e outros mais têm sido realizados com o objetivo de apoiar os engenheiros de ontologias na construção de ontologias, principalmente aplicando métodos que venham trazer agilidade e cada um deles cumpre o que se propõem dentro do escopo para o qual foi proposto. Existem muitas similaridades entre as abordagens e as ferramentas, mas também muitas diferenças que podem determinar uma modelagem mais consistente e eficiente. Assim, nossa intenção é mostrar o que de novo o OLED estendido pode contribuir com a comunidade.

Uma das áreas de pesquisa que pode se beneficiar do OLED estendido são as de criação de OPLs por terem no OLED estendido uma forma de criação e distribuição de catálogos de DROPs com um processo de seleção de DROPs. Nesse quesito, a abordagem de catálogos de DROPs com apoio automatizado materializada no OLED estendido, mostrou-se uma ferramenta prática e eficiente para a modelagem de ontologias de domínio.

8 Conclusões

O desenvolvimento de ontologias a partir do zero e/ou sem um especialista no domínio é uma tarefa árdua e muito complexa. Com o auxílio de ferramentas e abordagens metodológicas para a construção de ontologias, o trabalho se torna mais suave, mas o esforço pelo desconhecimento do domínio em que se constrói a ontologia ainda é um entrave. Assim, algumas formas de construção de ontologias foram buscadas, tanto no âmbito de ferramentas disponíveis quanto na construção de catálogo de DROPs. Depois de uma análise, o OLED foi escolhido como ferramenta de suporte junto com a abordagem de catálogo de padrões ontológicos como ferramentas para a implementação e utilização de catálogos DROPs. Para isso, foi criado um módulo para trabalhar os aspectos da criação e utilização de catálogos de DROPs de forma a não existir restrição quanto ao domínio do catálogo em implementação. Podem ser criados catálogos de DROPs, em quaisquer domínios tendo como base as categorias OntoUML implementadas pelo OLED estendido que ainda conta com funcionalidade para importação destes catálogos.

Uma tarefa desafiadora na modelagem de ontologia está relacionada a seleção das melhores ontologias para reutilização, daí estendê-las e compor com elas vários fragmentos de ontologia (GANGEMI; PRESUTTI, 2009). A abordagem metodológica apresentada nesta dissertação vem ao encontro dessa necessidade oferecendo um processo sistemático e definido para que DROPs possam ser extraídos de ontologias de referência e sejam organizados em catálogos de DROPs com o apoio de ferramenta computacional (módulo de gerenciamento de catálogo de DROPs para o OLED), contribuindo com um processo estruturado e automatizado favorece, com isso, a manutenção de ontologias e catálogos de DROPs.

A abordagem de catálogo de DROPs implementada no OLED estendido, sistematiza a construção de catálogos de DROPs e ontologias operacionais de domínio que podem ser compartilhadas e utilizadas na construção de outras ontologias. O processo de construção se tornou mais dinâmico, robusto e resistente a inconsistências uma vez que todo o ferramental já presente no OLED continua válido para os catálogos. A construção e utilização de catálogo de DROPs para a criação de ontologias operacionais ganharam a padronização que os DROPs oferecem tornando a construção de ontologias muito análoga a montar formas ou objetos utilizando LEGO[®].

Uma vez que o OLED implementa as categorias de OntoUML (UFO-A), a situação que traz muitas dificuldades aos engenheiros de ontologias que é de escolher uma linguagem de representação para modelagem não mais é objeto de preocupação, pelo fato de OntoUML ser uma linguagem formal, bem fundamentada e expressiva. Logo, se propicia o

compartilhamento do conhecimento de domínios dentro de uma comunidade removendo ou diminuindo a possibilidade que conceituações tenham interpretações diferentes em comunidades distintas levando a consistência e removendo ambiguidades no compartilhamento, o que é desejável. Isso acaba se refletindo na modelagem, quando utilizamos a abordagem de catálogos de DROPs, proposta neste trabalho, para a construção de ontologias.

Assim, três domínios foram escolhidos, serviço, processo de software baseado na ISO e colaboração, e o OLED estendido se mostrou flexível ao prover recursos necessários para reutilização de padrões por meio dos catálogos de DROPs que foram utilizados para a validação da abordagem e da ferramenta com a aplicação deles em alguns casos de estudos.

A abordagem de catálogos de DROPs com suporte automatizado fornecido pelo OLED estendido, se mostrou muito promissora e de grande proveito, explorando a reutilização de uma forma prática. Os catálogos de DROPs podem ser vistos como uma camada entre o conhecimento do domínio e uma ontologia operacional de domínio abstraindo do engenheiro de ontologias uma visão de mais baixo nível. A visão passa ser de padrões que lidam com os problemas de modelagem daquele domínio. Essa é uma visão de nível mais alto, uma visão mais holística da modelagem de ontologias.

8.1 Dificuldades

As principais dificuldades encontradas durante a realização deste trabalho foram:

- Seleção de uma ferramenta que permitisse a implementação da abordagem de catálogo de DROPs tendo em vista as restrições existentes;
- Falta de documentação da ferramenta escolhida (OLED) para entendimento das funcionalidades e módulos ali existentes, bem como a melhor forma de utilizá-los para alcançar os objetivos;
- Deficiência da ferramenta e módulos utilizados no que tange ao gerenciamento de memória, causando vários estouros de memória gerando retrabalhos. Dificuldade essa que foi contornada com o aumento de memória alocada para o programa, mas que esbarra na limitação física dela no computador, e diminuindo o tamanho da modelagem para modelagens parciais de ontologias e catálogos;
- Falta de implementação de outras categorias de OntoUML não pertencentes a UFO-A para a modelagem de ontologias. Isso foi contornado mapeando essas categorias para categorias presentes;
- Funcionalidade que não estavam funcionais. Problema contornado com a correção das que eram imprescindíveis para a continuidade do trabalho. As que não impactavam ou não eram tão necessárias não foram alteradas.

8.2 Contribuições

As principais contribuições que resultaram deste trabalho foram:

- Uma abordagem metodológica para a criação de catálogos de DROPs sistematizando e padronizando em etapas o processo, mostrando algumas orientações sobre como definir DROPs que possam ser reutilizados no desenvolvimento de ontologias de domínio (Capítulo 4);
- A criação de um catálogo de DROPs para o domínio de Colaboração sendo apresentados, mesmo que com uma especificação parcial dos padrões, com suas especificações e suas relações (Capítulo 5; Capítulo 3, na seção 3.3.2);
- A criação de catálogos de DROPs parciais para os domínios de Serviço (S-OPL) e de Processo de Software baseado na ISO (ISP-OPL) sendo utilizados, mesmo que com uma implementação parcial dos padrões, para a utilização na validação da abordagem implementada na ferramenta OLED (Capítulo 6, seções 6.1 e 6.2);
- A implementação da abordagem de catálogo de DROPs em uma ferramenta computacional (módulo de gerenciamento de catálogo de DROPs no OLED), por meio do gerenciamento deles, permitindo que catálogos de DROPs em quaisquer domínios possam ser criados e utilizados na modelagem de ontologias (Capítulo 5);
- A proposta e apresentação de uma versão preliminar de uma OPL para o domínio de colaboração (C-OPL) com a seu modelo do processo para a aplicações dos DROPs (Capítulo 4, seção 4.5);
- Fornecer apoio computacional para a reutilização de ontologias por meio de catálogos de DROPs oferecendo um processo sistemático e definido para que DROPs possam ser extraídos de ontologias de referência e sejam organizados em catálogos de DROPs para posterior utilização;

8.3 Trabalhos Futuros

Considerando o estágio atual deste trabalho, alguns dos possíveis desdobramentos dele, gerando novos trabalhos que podem ser realizados no futuro, são apresentados a seguir:

- Evoluir a CONTO por meio de reavaliações iterativas quanto a abrangência e a qualidade dos conceitos ali modelados para que ela esteja alinhada com o maior número de soluções de problemas recorrentes no domínio de colaboração;

- Reavaliar os DROPs de colaboração quanto a solução de problemas recorrentes de modelagem, dimensionando os mesmos para que sejam atendidos os requisitos com o menor grupo de conceitos e DROPs;
- Aprimorar a abordagem metodológica proposta na capítulo 4 para que seja a mais simples e prática possível no auxílio a criação de catálogos de DROPs de forma a sistematizar o trabalho e padronizando-o;
- Incrementar o módulo de gerenciamento de catálogo de DROPs para que se torne cada vez mais funcional como ferramenta de apoio com a inclusão de novas funcionalidades e aprimoramento da existentes;
- Criar catálogos de DROPs, inclusive em outros domínios, para utilização na modelagem de ontologias para validação da abordagem e implementação dela na ferramenta;
- Criar mecanismos com que se possam realizar todas as etapas do processo de criação de OPL, permitindo, de forma interativa, a criação do modelo de processo, dos padrões que formam o catálogo da OPL, da configuração dos padrões do catálogo, partindo de uma ontologias de referência;
- Gerar a(s) ontologia(s) de referência que estão associadas ao catálogo de DROPs e/ou a OPL;
- Exportar e/ou disponibilizar a documentação de especificação de catálogos e de OPLs para que possa ser acessada e utilizada para a modelagem de ontologias;
- Criar mecanismos de apoio a modelagem em que a ferramenta sugira, ao engenheiro de ontologias, OPLs, catálogos de DROPs e DROPs, tornando o processo menos dependente do modelador;
- Criar metodologias que apoiem a automatização do processo de modelagem de ontologias utilizando catálogos de DROPs e OPLs;
- Criar repositórios de catálogo de DROPs para a utilização na ferramenta para que possam ser utilizados e construídos de forma colaborativa, bem como de OPLs.

Referências

- ALBUQUERQUE, A.; GUIZZARDI, G. An ontological foundation for conceptual modeling datatypes based on semantic reference spaces. In: IEEE. *Research Challenges in Information Science (RCIS), 2013 IEEE Seventh International Conference on*. 2013. p. 1–12. Disponível em: <<http://ieeexplore.ieee.org/document/6577693/>>. Citado na página 36.
- ARPÍREZ, J. et al. Webode: a scalable ontological engineering workbench. In: *First International Conference on Knowledge Capture (K-CAP 2001)*. Victoria, Canada. [s.n.], 2001. Disponível em: <<http://oa.upm.es/5483/1/KCAP01-pdf-ACM.pdf>>. Citado na página 144.
- BARCELOS, P. P. F. et al. An automated transformation from ontouml to owl and swrl. In: *Ontobras*. [s.n.], 2013. p. 130–141. Disponível em: <http://ceur-ws.org/Vol-1041/ontobras-2013_paper44.pdf>. Citado na página 37.
- BLOMQVIST, E. *Semi-automatic ontology construction based on patterns*. Tese (Doutorado) — Linköpings universitet, 2009. Acessado em 15/10/2016. Disponível em: <<http://liu.diva-portal.org/smash/get/diva2:207543/FULLTEXT01.pdf>>. Citado 2 vezes nas páginas 23 e 40.
- BLOMQVIST, E.; GANGEMI, A.; PRESUTTI, V. Experiments on pattern-based ontology design. In: ACM. *Proceedings of the fifth international conference on Knowledge capture*. [S.l.], 2009. p. 41–48. Citado na página 73.
- BORGHOFF, U. M.; SCHLICHTER, J. H. *Computer-Supported Cooperative Work: Introduction to Distributed Applications*. [S.l.]: Springer Berlin Heidelberg, 2013. ISBN 9783662042328. Citado na página 61.
- BUSCHMANN, F.; HENNEY, K.; SCHMIDT, D. C. *Pattern-oriented software architecture, on patterns and pattern languages*. [S.l.]: John Wiley & Sons Ltd, 2007. v. 5. 8–8 p. Citado 4 vezes nas páginas 41, 44, 92 e 109.
- CAMPOS, M. L. de A. et al. Ontologias: representando a pesquisa na área através de mapa conceitual. 2012. Disponível em: <<http://enancib.ibict.br/index.php/enancib/viiienancib/paper/viewFile/2827/1955>>. Citado 3 vezes nas páginas 30, 31 e 32.
- CARRARETTO, R. A modeling infrastructure for ontouml. *Graduation Thesis, Federal University of Espírito Santo*, 2010. Disponível em: <https://nemo.inf.ufes.br/wp-content/papercite-data/pdf/a_modeling_infrastructure_for_ontouml_2010.pdf>. Citado na página 36.
- CMMI, C. *OntoUML*. 2016. Acessado em 06/10/2016. Disponível em: <<http://ccmi.fit.cvut.cz/methodologies/ontouml/>>. Citado 2 vezes nas páginas 35 e 36.
- D'AQUIN, M. et al. Neon tool support for building ontologies by reuse. In: DEMO, INTERNATIONAL CONFERENCE ON BIOMEDICAL ONTOLOGY, ICBO. 2009. Disponível em: <<https://pdfs.semanticscholar.org/e6bf/25c132d5880a2be2047faec3a74b29c20c84.pdf>>. Citado na página 143.

- ERIKSSON, H. et al. Task modeling with reusable problem-solving methods. *Artificial Intelligence*, Elsevier, v. 79, n. 2, p. 293–326, 1995. Citado na página 145.
- FALBO, R. A. Sabio: Systematic approach for building ontologies. In: *ONTO. COM/ODISE@ FOIS*. [s.n.], 2011. Disponível em: <<https://pdfs.semanticscholar.org/2660/676b0d784d46d3c407793c0cee0c10e76c5b.pdf>>. Citado 4 vezes nas páginas 24, 32, 74 e 92.
- FALBO, R. A. et al. Organizing ontology design patterns as ontology pattern languages. In: . [s.n.], 2013. Springer. pp 61–75. Disponível em: <<https://eswc-conferences.org/sites/default/files/papers2013/falbo.pdf>>. Citado 4 vezes nas páginas 22, 24, 41 e 43.
- FALBO, R. A. et al. Ontology patterns: clarifying concepts and terminology. In: CEUR-WS. ORG. *Proceedings of the 4th International Conference on Ontology and Semantic Web Patterns-Volume 1188*. 2013. p. 14–26. Disponível em: <http://ceur-ws.org/Vol-1188/paper_11.pdf>. Citado 5 vezes nas páginas 9, 21, 41, 42 e 109.
- FALBO, R. A. et al. S-opl: Service ontology pattern language specification version 1.5. 2015. Disponível em: <<https://nemo.inf.ufes.br/wp-content/uploads/2015/12/s-opl-v1.5.pdf>>. Citado 10 vezes nas páginas 9, 13, 26, 43, 44, 45, 46, 47, 48 e 49.
- FALBO, R. A. et al. Isp-opl – iso-based software process opl. In: . [s.n.], 2015. Disponível em: <https://nemo.inf.ufes.br/wp-content/uploads/2015/03/ISP-OPL_Specification_v1.0.pdf>. Citado 19 vezes nas páginas 9, 10, 13, 26, 44, 45, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 116 e 118.
- FALBO, R. A. et al. An ontology pattern language for service modeling. In: ACM. *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. 2016. p. 321–326. Disponível em: <https://nemo.inf.ufes.br/wp-content/uploads/2015/03/An-ontology-pattern-language-for-service-modeling_2016.pdf>. Citado 10 vezes nas páginas 11, 26, 41, 45, 46, 47, 78, 110, 114 e 115.
- FALBO, R. A. et al. Towards an enterprise ontology pattern language. In: ACM. *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. 2014. p. 323–330. Disponível em: <<https://inf.ufes.br/~gguizzardi/OPL-SAC2014.pdf>>. Citado 3 vezes nas páginas 22, 36 e 51.
- FERNÁNDEZ-LÓPEZ, M.; GÓMEZ-PÉREZ, A.; GUARINO, N. *The methontology & ontoClean merge*. [S.l.], 2001. Citado na página 145.
- GAILLY, F.; ROELEN, B.; GUIZZARDI, G. The design of a core value ontology using ontology patterns. In: SPRINGER. *International Conference on Conceptual Modeling*. 2016. p. 183–193. Disponível em: <<https://www.inf.ufes.br/~gguizzardi/QMMQ.pdf>>. Citado na página 42.
- GANGEMI, A.; PRESUTTI, V. Ontology design patterns. In: *Handbook on ontologies*. [S.l.]: Springer, 2009. p. 221–243. Citado 2 vezes nas páginas 23 e 147.
- GASEVIC, D.; DJURIC, D.; DEVEDZIC, V. *Model Driven Engineering and Ontology Development*. [S.l.]: Springer Science Business Media, 2009. ISBN 9783642002816. Citado 3 vezes nas páginas 9, 30 e 31.

GRAÇAS, A. P.; GUIZZARDI, G. Padrões de modelagem e regras de construção de modelos para a criação de ontologias de domínio bem-fundamentadas em ontouml. 2010. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/ontobras/2010/0030.pdf>>. Citado na página 35.

GRAY, B. Collaborating: Finding common ground for multiparty problems. Jossey-Bass San Francisco, CA, 1989. Disponível em: <https://www.jstor.org/stable/258026?seq=1#page_scan_tab_contents>. Citado na página 122.

GRUBER, T. R. *A Translation Approach to Portable Ontology Specifications*. Stanford University - Computer Science Department, 1993. Disponível em: <<http://tomgruber.org/writing/ontolingua-kaj-1993.pdf>>. Citado 2 vezes nas páginas 23 e 29.

GUARINO, N. Formal ontology, conceptual analysis and knowledge representation. *Juornal*, 1995. Citado na página 29.

GUARINO, N. Understanding, building and using ontologies. *International Journal of Human-Computer Studies*, Elsevier, v. 46, n. 2-3, p. 293–310, 1997. Citado na página 30.

GUARINO, N.; WELTY, C. Evaluating ontological decisions with ontoclean. *Communications of the ACM*, ACM, v. 45, n. 2, p. 61–65, 2002. Citado na página 145.

GUERSON, J.; ALMEIDA, J. P.; GUIZZARDI, G. Support for domain constraints in the validation of ontologically well-founded conceptual models. In: *Enterprise, Business-Process and Information Systems Modeling*. Springer, 2014. p. 302–316. Disponível em: <https://nemo.inf.ufes.br/wp-content/papercite-data/pdf/support_for_domain_constraints_in_the_validation_of_ontologically_well_founded_conceptual_models_2014.pdf>. Citado na página 37.

GUERSON, J. et al. Ontouml lightweight editor: A model-based environment to build, evaluate and implement reference ontologies. In: IEEE. *2015 IEEE 19th International Enterprise Distributed Object Computing Workshop*. 2015. p. 144–147. Acessado em 15/10/2016. Disponível em: <<http://ieeexplore.ieee.org/document/7310684/>>. Citado 4 vezes nas páginas 9, 37, 38 e 39.

GUIZZARDI, G. *Ontological Foundations for Structural Conceptual Models*. Tese (Doutorado) — University of Twente, 2005. Acessado em 15/10/2016. Disponível em: <<http://www.inf.ufes.br/~gguizzardi/OFSCM.pdf>>. Citado 6 vezes nas páginas 9, 29, 33, 34, 35 e 74.

GUIZZARDI, G. On ontology, ontologies, conceptualizations, modeling languages, and (meta) models. *Frontiers in artificial intelligence and applications*, IOS Press, v. 155, p. 18, 2007. Citado na página 33.

HAASE, P. et al. The neon ontology engineering toolkit. WWW, 2008. Disponível em: <http://s3.amazonaws.com/academia.edu.documents/30706200/10.1.1.141.4163.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1496725132&Signature=P2GsGY8J5l6YulaCkiYV7kgCnaw%3D&response-content-disposition=inline%3B%20filename%3DThe_neon_ontology_engineering_toolkit.pdf>. Citado na página 143.

HACKING, I. *Historical Ontology*. [S.l.]: Harvard University Press, 2004. ISBN 9780674016071. Citado na página 29.

HENDLER, J. Agents and the semantic web. IEEE Educational Activities Department Piscataway, 2001. Citado na página 30.

ISOTANI, S.; BITTENCOURT, I. I. *Dados Abertos Conectados: Em busca da Web do Conhecimento*. [S.l.]: Novatec Editora, 2015. Citado 5 vezes nas páginas 9, 21, 30, 31 e 32.

KALFOGLOU, Y. Exploring ontologies. 2001. Citado na página 30.

KNOLL, S. W. et al. Collaboration ontology: Applying collaboration knowledge to a generic group support system. In: *Group Decision and Negotiation Meeting (GDN) 2010*. [s.n.], 2010. p. 12–26. Disponível em: <https://www.dai-labor.de/fileadmin/Files/Publikationen/Buchdatei/Collaboration%20Ontology%20Applying%20Collaboration%20Knowledge%20to%20a%20Generic%20Group%20Support%20System_gdn10_knoll.pdf>. Citado na página 122.

LLOYD, S. Quantum enigma machines. *arXiv preprint arXiv:1307.0380*, 2013. Citado na página 40.

LÓPEZ, M. F. et al. Building a chemical ontology using methontology and the ontology design environment. *IEEE Intelligent Systems and their applications*, IEEE, v. 14, n. 1, p. 37–46, 1999. Disponível em: <http://oa.upm.es/5466/1/Building_a_Chemical_Ontology.pdf>. Citado na página 144.

MOLINA, M. et al. A structure of problem-solving methods for real-time decision support: Modeling approaches using protégé-ii and ksm. In: *Proc. Of Knowledge Acquisition of Knowledge Based Systems Workshop, KAW96. Banff, Canada*. [S.l.: s.n.], 1996. Citado 2 vezes nas páginas 145 e 146.

MUSEN, M. A. et al. Protege-ii: computer support for development of intelligent systems from libraries of components. *Medinfo*, v. 8, n. Pt 1, p. 766–770, 1995. Citado 2 vezes nas páginas 145 e 146.

NARDI, J. C. et al. A commitment-based reference ontology for services. *Information systems*, Elsevier, v. 54, p. 263–288, 2015. Disponível em: <https://www.researchgate.net/profile/Giancarlo_Guizzardi2/publication/274700112_A_Commitment-based_Reference_Ontology_for_Services/links/558878df08ae8c4f3406493f.pdf>. Citado 2 vezes nas páginas 46 e 47.

NEMO, O. . C. M. R. G. Oled - ontouml lightweight editor. In: . [s.n.], 2016. Acessado em 15/10/2016. Disponível em: <<http://nemo.inf.ufes.br/projects/oled/>>. Citado na página 37.

OLIVEIRA, F. F. *Uma Ontologia de Colaboração e suas aplicações*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, 2009. Acessado em 12/10/2016. Disponível em: <http://nemo.inf.ufes.br/wp-content/papercite-data/pdf/uma_ontologia_de_colaboracao_e_suas_aplicacoes_2009.pdf>. Citado 14 vezes nas páginas 10, 13, 25, 26, 45, 60, 61, 62, 63, 64, 65, 66, 74 e 122.

OLIVEIRA, F. F. Uma ontologia de colaboração e suas aplicações. In: . [S.l.: s.n.], 2009. (in Portuguese) Masters Dissertation, Universidade Federal do Espírito Santo. Available at: <http://portais4.ufes.br/posgrad/teses/tese335.pdf>. Citadonapágina25.

PIMENTEL, M. *Uma Ontologia de Colaboração e suas Aplicações*. Tese (Doutorado)

- UFES, 2006. Acessado em 12/10/2016. Disponível em: <https://nemo.inf.ufes.br/wp-content/papercite-data/pdf/uma_ontologia_de_colaboracao_e_suas_aplicacoes_2009.pdf>. Citado na página 61.
- POIDEVIN, R. L. et al. *The Routledge Companion to Metaphysics*. [S.l.]: Routledge, 2009. ISBN 9780415493963. Citado na página 29.
- PROTÉGÉ, S. C. for B. I. R. Protégé. In: . [s.n.], 2017. Acessado em 23/05/2017. Disponível em: <<http://protege.stanford.edu/products.php>>. Citado na página 146.
- QUIRINO, G. K. S. *Notação Visual para Representação de Linguagens de Padrões Ontológicos*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, 2016. Acessado em 12/12/2016. Disponível em: <http://repositorio.ufes.br/bitstream/10/4304/1/tese_9686_DissertacaoGlaiceKellyQuirinoVF.pdf>. Citado 3 vezes nas páginas 21, 41 e 78.
- QUIRINO, G. K. S. et al. Towards a service ontology pattern language. In: SPRINGER. *International Conference on Conceptual Modeling*. 2015. p. 187–195. Disponível em: <https://nemo.inf.ufes.br/wp-content/uploads/2015/12/towards_a_service_ontology_pattern_language_2015.pdf>. Citado 9 vezes nas páginas 11, 26, 36, 46, 47, 109, 110, 111 e 116.
- RUY, F. B. et al. An iso-based software process ontology pattern language and its application for harmonizing standards. *ACM SIGAPP Applied Computing Review*, ACM, v. 15, n. 2, p. 27–40, 2015. Disponível em: <<http://www.inf.ufes.br/~gguizzardi/SigApp2015.pdf>>. Citado 5 vezes nas páginas 11, 26, 50, 116 e 118.
- RUY, F. B. et al. Towards an ontology pattern language for harmonizing software process related iso standards. In: ACM. *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. 2015. p. 388–395. Disponível em: <<https://pdfs.semanticscholar.org/c8ab/582f811b682091b48055978cf90f074f95f6.pdf>>. Citado 2 vezes nas páginas 11 e 117.
- RUY, F. B. et al. Towards an ontology pattern language for harmonizing software process related iso standards. In: ACM. *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. 2015. p. 388–395. Disponível em: <https://nemo.inf.ufes.br/wp-content/papercite-data/pdf/towards_an_ontology_pattern_language_for_harmonizing_software_process_related_iso_standards_2015.pdf>. Citado na página 33.
- RUY, F. B. et al. Seon: A software engineering ontology network. In: *EKAW*. [s.n.], 2016. p. 527–542. Disponível em: <https://nemo.inf.ufes.br/wp-content/uploads/2016/10/SEON_A-Software-Engineering-Ontology-Network-Ruy-et-al.-2016.pdf>. Citado na página 116.
- RUY, F. B. et al. Ontology engineering by combining ontology patterns. In: SPRINGER. *International Conference on Conceptual Modeling*. 2015. p. 173–186. Disponível em: <https://nemo.inf.ufes.br/wp-content/papercite-data/pdf/ontology_engineering_by_combining_ontology_patterns_2015.pdf>. Citado 3 vezes nas páginas 9, 33 e 42.
- SALES, T. P. *Ontology validation for managers*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, 2014. Acessado em 15/10/2016. Disponível em:

- https://nemo.inf.ufes.br/wp-content/papercite-data/pdf/ontology_validation_for_managers_2014.pdf>. Citado na página 37.
- SILVA, A. R. L. et al. Ontologia como representação do conhecimento: aplicação no curso de formação continuada em tecnologias educacionais na web. *RENOTE*, 2014. Citado na página 32.
- SILVA, E. O. *Integração de padrões de análise e ontologias de domínio: um estudo de caso no domínio de gestão urbana*. Dissertação (Mestrado) — Universidade Federal de Viçosa, 2008. Acessado em 27/10/2016. Disponível em: <<http://www.locus.ufv.br/bitstream/handle/123456789/2584/texto%20completo.pdf?sequence=1&isAllowed=y>>. Citado na página 45.
- SOUZA, A. K. et al. Extending an ontology editor for domain-related ontology patterns reuse: An application in the collaboration domain. *Juornal*, October 2016. Citado 12 vezes nas páginas 10, 11, 14, 21, 22, 25, 26, 45, 61, 77, 122 e 125.
- SWARTOUT, W.; TATE, A. Ontologies. *IEEE Intelligent Systems and their Applications*, 1999. Citado na página 30.
- TERRA, J. C. C. et al. *Portais corporativos: a revolução na gestão do conhecimento*. [S.l.]: Negócio Editora, 2002. Citado na página 61.
- TERVEEN, L. G. Overview of human-computer collaboration. *Knowledge-Based Systems*, Elsevier, v. 8, n. 2-3, p. 67–81, 1995. Citado na página 122.
- VALASKI, J.; REINEHR, S.; MALUCELLI, A. Computational environment to semi-automatically build a conceptual model represented in ontouml. 2016. Disponível em: <https://www.researchgate.net/publication/310004927_Computational_Environment_to_Semi-Automatically_Build_a_Conceptual_Model_Represented_in_OntoUML>. Citado na página 23.
- WEBSTER, M. *Merriam Webster Dictionary*. 2016. Acessado em 15/10/2016. Disponível em: <<http://www.merriam-webster.com/>>. Citado 2 vezes nas páginas 60 e 122.
- WELTY, C.; ANDERSEN, W. Towards ontoclean 2.0: A framework for rigidity. *Applied Ontology*, IOS Press, v. 1, n. 1, p. 107–116, 2005. Citado na página 145.
- ZAMBORLINI, V. C. *Estudo de Alternativas de Mapeamento de Ontologias da Linguagem OntoUML Para OWL: Abordagens Para Representação de Informação Temporal*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, 2011. Acessado em 27/10/2016. Disponível em: <http://repositorio.ufes.br/bitstream/10/4238/1/tese_4171.pdf>. Citado na página 35.
- ZHDANOVA, A. V.; KELLER, U. Choosing an ontology language. In: *WEC (2)*. [s.n.], 2005. p. 47–50. Disponível em: <https://www.researchgate.net/profile/Anna_Fensel/publication/221017661_Choosing_an_Ontology_Language/links/00b4953767201e5afd000000/Choosing-an-Ontology-Language.pdf?origin=publication_detail>. Citado na página 23.

Apêndices

APÊNDICE A – Documento de Especificação do Catálogo de DROPs de Colaboração

C-OPL: Collaboration Ontology Pattern Language

Specification Version 1.0

André K. Souza
Renata S. S. Guizzardi
Maria L. M. Campos

July, 2017

Summary

1. Introduction.....	4
2. C-OPL Domain-Related Ontology Patterns	5
3. C-OPL Process	7
4. C-OPL Patterns Descriptions	14
4.1 <i>Cooperation Group</i>	15
CSL – Collaborative Session Location.....	15
LT – Location Type.....	15
CSAC – Collaborative Session Action Contributions	16
CSAD – Collaborative Session Agreement Description	16
CSA – Collaborative Session Agents	17
CA – Collaborative Agreement	17
CAD – Collaborative Agreement Description.....	18
CAC – Collaborative Agreement Composition.....	18
CCAC – Collaborative Commitment for Action Contributions.....	19
CCIC – Collaborative Commitment Inheres in Collaborator	19
ACC – Acton Contribution Composition	20
ACD – Action Contribution Dependence	20
ACPD – Action Contribution Plan Description	21
ACPPS – Action Contribution Pre and Pos Situation.....	21
APAC – Agent Performs Action Contribution.....	22
GPCCC – Goal as the Propositional Content of a Collaborative Commitment	22
GC – Goal Conflict	23
GD – Goal Dependence	23
4.2 <i>Coordination Group</i>	24
CSRS - Collaborative Session Resource Simple	24
CSR - Collaborative Session Resource.....	24
RC - Resource Composition.....	25
RN - Resource Nature.....	25
4.3 <i>Communication Group</i>	26
CIPS – Communication Interaction Pattern Simple	26
CIP – Communication Interaction Pattern.....	26
IL – Idiomatic Language	27
5. References	28

C-OPL: Collaboration Ontology Pattern Language

This document presents the Collaboration Ontology Pattern Language (C-OPL). In this version of C-OPL specification (1.0), C-OPL is represented by using OPL-ML (Ontology Pattern Language Modeling Language). Thus, a structured model is used to represent the C-OPL patterns and structural relationships between them, a general process model provides a general view of the C-OPL process, and detailed process models expand the process general view.

1. Introduction

An **Ontology Pattern Language** (OPL) aims to provide holistic support for using Domain-Related Ontology Patterns (DROPs) in the development of a domain ontology. It provides explicit guidance on what modeling problems can arise in that domain, informs the order to address these problems, and suggests one or more patterns to solve each specific problem. Moreover, an OPL explicitly considers pattern combinations to solve a given problem (as well as conflicting patterns), along with guidelines for integrating them into a concrete ontology conceptual model (FALBO et al., 2013).

An OPL indicates explicitly which patterns must be considered as mandatory and which ones as optional. To ensure a stable and sound pattern usage, the order in which such patterns are combined is structured by means of a *pattern application flow* (the OPL process). OPLs are structured to support and encourage the application of one pattern at a time. Step by step, the patterns are chosen and combined in a systematic way. This ensures that the main property of piecemeal growth is preserved: the ‘whole’ always precedes its ‘parts’ (FALBO et al., 2013).

Comply with OLIVEIRA et al. (2007), the reuse of knowledge is a determining factor in organizations such as, for example, the existence of an organizational memory. Thus, the ability to reuse knowledge can make organizations faster in response to the various situations dealt with and performed by them (OLEARY, 1998). In this sense, the OPLs, with their respective process models, facilitate modeling by providing a set of DROPs with fragments of domain knowledge specialized in solving a recurring modeling problem.

The Collaboration Ontology Pattern Language (C-OPL) is an OPL that addresses the core conceptualization about the collaboration phenomena. C-OPL patterns were extracted from CONTO, a reference ontology for collaboration (OLIVEIRA, 2009), that formalize knowledge about the collaboration domain in order to provide a common vocabulary, promoting integration within the collaboration domain. CONTO was proposed as the means to integrate several collaborative software applications aiming at effectively collaboration support within organizations (OLIVEIRA et al., 2007) and is based on the structure defined by the 3C Model (*Communication, Coordination and Cooperation*) (ELLIS et al., 1991).

The patterns that compose C-OPL are briefly presented in Section 2. C-OPL structural model is presented in Section 3. C-OPL process model is described in Section 4. Finally, in the Section 5, each C-OPL pattern is described.

2. C-OPL Domain-Related Ontology Patterns

As aforementioned, the patterns that compose C-OPL are organized in three groups, namely: (i) *Cooperation*, (ii) *Coordination*, and (iii) *Communication*.

The backbone of cooperation can be summed up in "who does what where". All other cooperation concerns derive from this. The "who" is represented by the cooperation's *agent*, the "does what" is presented by the *actions contribution* that the agent performs and "where" is represented by the *collaborative sessions* in which the *agents* participate. Thus, *collaborative sessions* happen in *sites* which can be *real sites* or *virtual sites*.

The *collaborative sessions* are ruled by *collaborative agreements* that can be decomposed into *atomic collaborative agreements*, when they can not be decomposed into others *atomic collaborative agreements*, and into *complex collaborative agreements*, when they can be decomposed into other *complex collaborative agreements* or *atomic collaborative agreements*. These *collaboration agreements* are described in the *collaboration agreements description* that are used by the *collaborative sessions*.

The *collaborative sessions* are essentially composed of the *actions contribution* carried out by the *agents* who participate in these sessions. These *actions contribution* may depend on other *actions contribution* and may lead to *situations* before and after their respective executions. They are also described in *plans descriptions* and are caused by *collaborative commitments*. The *collaboration commitments* are inherent to *collaborators* and have *goals* as propositional content. These *collaboration goals*, in turn, may depend on other *goals* and may conflict with other *goals*. Table 1 describes the intent of the patterns of the Cooperation group.

Table 1 - Patterns of the Cooperation group.

Id	Name	Intent
CSL	Collaborative Session Location	To represent the site where the collaborative session happens
LT	Location Type	To represent the type of site where the collaborative session happens
CSAC	Collaborative Session Action Contributions	To represent the action contributions that compose the collaborative session
CSAD	Collaborative Session Agreement Description	To represent the descriptions of the collaborative session
CSA	Collaborative Session Agents	To represent the agents that participate in the collaborative session
CA	Collaborative Agreement	To represent the collaborative agreement that rules the collaborative session
CAD	Collaborative Agreement Description	To represent the description of the collaborative agreements

CAC	Collaborative Agreement Composition	To represent collaborative agreement mereological decomposition
CCAC	Collaborative Commitment for Action Contributions	To represent the collaborative commitment that causes the action contribution
CCIC	Collaborative Commitment Inheres in Collaborator	To represent the collaborative commitment that inheres in a collaborator
ACC	Action Contribution Composition	To represent action contribution mereological decomposition
ACD	Action Contribution Dependence	To represent the dependence between two action contributions
ACPD	Action Contribution Plan Description	To represent the plan description of the action contributions
ACPPS	Action Contribution Pre and Pos Situation	To represent the pre and pos situation of the action contributions
APAC	Agent Performs Action Contribution	To represent the collaborative roles instantiated by the agents that participate in the collaborative session
GPCCC	Goal as the Propositional Content of a Collaborative Commitment	To represent the goal that is the propositional content of a collaborative commitment
GC	Goal Conflict	To represent the conflict between two goals
GD	Goal Dependence	To represent the dependence between two goals

The coordination deals with the participation of *resources* in *collaborative sessions*. They can be done in a simple way, *collaborative session resource simple*, or can be done in a more detailed way, *collaborative session resource*, in which participation is mediated and materialized, *resource participation*, in the *collaborative session*. *Resources* can be specialized into *simple resources*, when they can not be decomposed into other simplest resource, and into *composite resources*, when they can be decomposed into other *composite resources* or *simple resources*. These *resources* can also be *exclusive resources*, when they can not be shared or can not be used simultaneously, or *shareable resources*, otherwise. Table 2 describes the intent of the patterns of the Coordination group.

Table 2 - Patterns of the Coordination group.

Id	Name	Intent
CSRS	Collaborative Session Resources Simple	To represent the resources that participate in the collaborative session
CSR	Collaborative Session Resource	To represent the participation of resources in collaborative session (with the mediation of a resource)

		participation relator)
RC	Resource Composition	To represent resource mereological decomposition
RN	Resource Nature	To represent resource types in terms of shareability

The communication between the *sender* and the *receiver* is mediated by the *communication interaction* that is the heart of communication. The *communication interactions* occur during a *collaborative session*, when participants exchanges information. The *communication interaction* is essentially composed of the *sender's communicative acts* and the *receiver's perceptions* of them. *Perceptions* are caused by *communicative acts*. *Messages* are always propositional content of *communicative acts*, *perceptions* and *communicative interactions*. These *messages* are coded in an *idiomatic language*. Table 3 describes the intent of the patterns of the Communication group.

Table 3 - Patterns of the Communication group.

Id	Name	Intent
CIP	Communication Interaction Pattern	To represent in detail an interaction pattern connecting a sender and a receiver
CIPS	Communication Interaction Pattern Simple	To represent an interaction pattern connecting a sender and a receiver
IL	Idiomatic Language	To represent the idiomatic language codifying the message

3. C-OPL Process

The figures 1-5 present the C-OPL process. Figure 1 provides a general view of the C-OPL process. Figures 2-4, in turn, show the detailed process related, respectively, to the Cooperation, Coordination and Communication groups. Finally, Figure 5 presents the C-OPL process in a complete and detailed view.

In Figures 2-5, actions concerning a pattern application are represented by *pattern application action nodes* (the labeled rounded rectangles). A *pattern application action* refers to the application of a specific pattern. In these figures, *pattern application action groups* are delimited by rectangles with blue edges and are showed in an expanded way, being possible to see the grouped pattern application actions. Moreover, *variant pattern application actions* are grouped inside rectangles with red dotted edges. In Figure 1, *pattern application action groups* are represented as black boxes, providing a more general view of C-OPL. In this figure, *pattern application action groups* are represented by labeled rectangles with blue edges and with the symbol like a small inverted trident in the bottom right corner.

Initial nodes (solid circles) are used to represent entry points in the OPL, i.e., pattern application actions in the language that can be performed first, without performing other pattern application actions. *Fork nodes* (line segments with multiple output flows) are used to represent parallel paths, i.e., if the ontology engineer decides to follow the fork node input path, then s/he must follow all paths leaving them. *Join nodes* (line segments with multiple input flows) are used to represent the junction of parallel paths and, to follow the join node output path, the ontology engineer must have already traveled all the join node input paths.

Decision nodes (represented by diamonds) are used to represent alternative paths. Thus, if the ontology engineer decides to follow the decision node input path, then s/he must select one and only one of the decision node output paths. *Control flows* (arrowed lines) represent the sequences of paths that the ontology engineer can follow in the OPL. *End points* (solid circle doubly circled) are used to indicate where the patterns application process can be finished. In the process model, different colors are used to identify application actions patterns from different groups.

The starting point for using an OPL is through the entry points of the language process model. The C-OPL has five entry points (EP1, EP2, EP3, EP4 and EP5) as Figure 5 shows. The ontology engineer must choose one, depending on the scope of the developing ontology. EP1 should be chosen when the requirements for the new ontology include the definition of where the collaboration will take place. EP2 must be chosen when the goal is to specify the actions contributions regardless of where they occur (EP1 is who cares with it).

EP3 is the option for when the interest is in the collaboration commitments without regard to the contribution actions (EP2) and where they occur (EP1). EP4 should be chosen when the requirements include coordination in the collaborative process on resource types and modeling the participation of these resources in the collaborative section. Finally, EP5 when the requirements include the definition of the communication during the collaboration without interest in how the cooperation will proceed and how the coordination will be done. EP1, EP2 and EP3 are entry points when the objective for the new ontology includes cooperation; EP4, if they include coordination and EP5, when they include communication without cooperation.

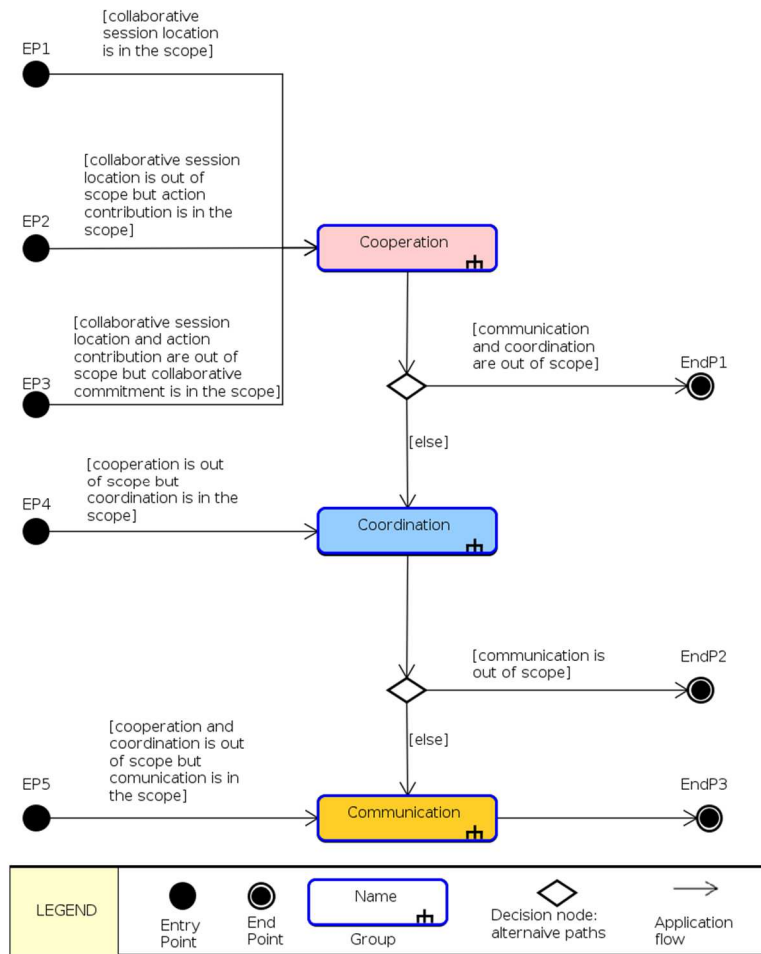


Figura 1 - General View of the C-OPL Process.

When EP1 is chosen, the ontology engineer must use first the *Collaborative Session Location* and the *Location Type* patterns, that belongs a patterns group, for modeling where the collaboration occurs. If the agent participation in the collaborative session is relevant, the *Collaborative Session Agent* pattern is select. After this, if the collaborative agreements are relevant for modeling, the *Collaborative Agreement* pattern can be selected. If more detailed information for the agreement is necessary, *Collaborative Agreement Description* and *Collaborative Session Agreement Description* are selected.

If the composition of the comlaborative agreement is relevant, the *Collaborative Agreement Composition* can be chosen. Once the collaborative session is modeled and the collaborative commitment, action contribution, coordination and communication are out of scope, the pattern application process finishes at the end point EndP1. Otherwise, the flow follows to action contribution or collaborative commitment patterns and/or the coordination and/or communication following the same way as when choosing the entry point EP4 and/or EP5.

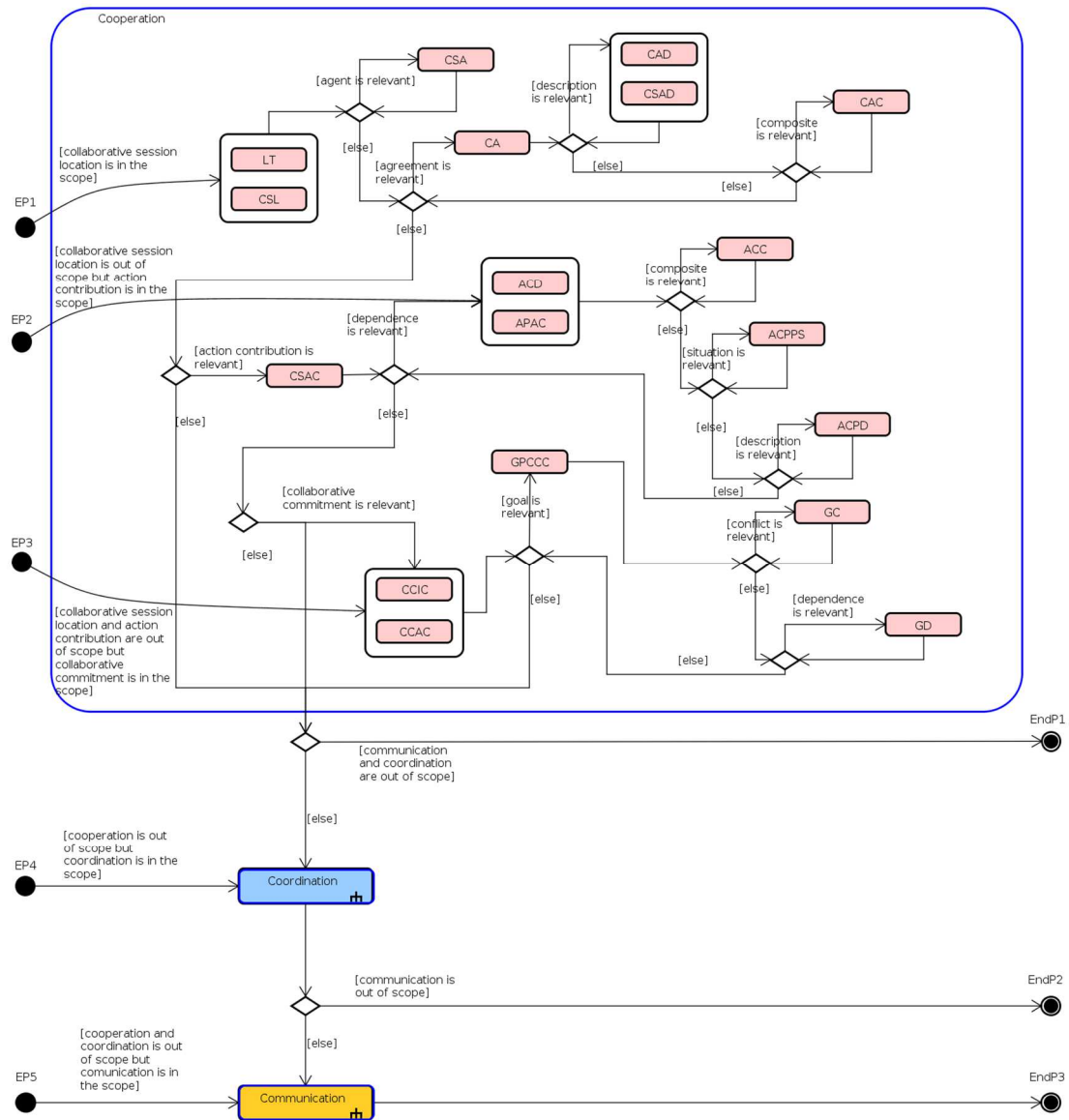


Figure 2 - Detailed View of the Cooperation Group.

If the collaborative session location is out of scope but the action contributions is in the scope, EP2 is chosen. Then, the *Collaborative Session Action Contribution* pattern is selected. If the dependences of the actions are relevant, the *Action Contribution* and *Agents Performs Action Contribution* patterns are selected. After this, the ontology engineer can choose the pattern *Action Contribution Composition*, if the composition of the action contribution is relevant to modeling. Other pattern that can be selected is *Action Contribution Pre and Pos Situation*, if the situations provoked by the contribution actions are important for the modeling.

Then next pattern that can be selected is the *Action Contribution Plan Description* to deal with the description of the action contributions. Once the action contribution is modeled and the collaborative commitment, the coordination and the communication is out of scope, the pattern application process finishes at the end point EndP1. Otherwise, the flow follows to collaborative commitment patterns and/or the

coordination and/or communication following the same way as when choosing the entry point EP4 and/or EP5.

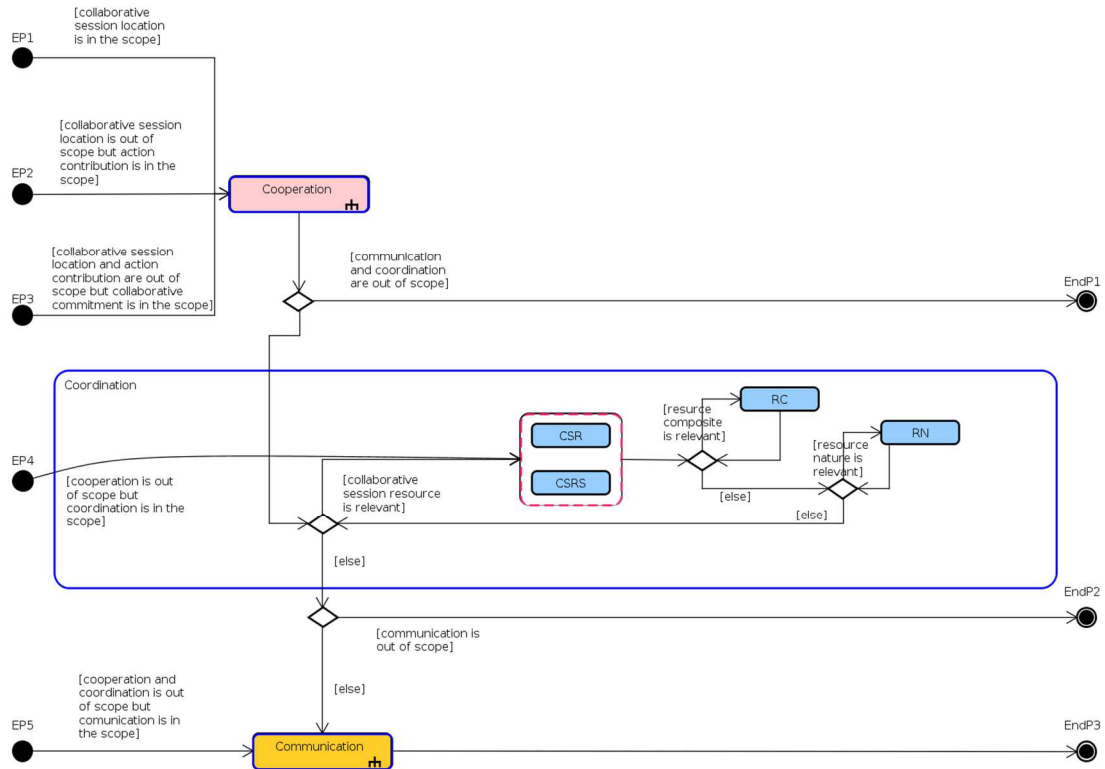


Figure 3 - Detailed View of the Coordination Group.

When the collaborative session location and the action contributions are out of scope but the collaborative commitment is in the scope, EP3 is chosen. Then, the ontology engineer must use first the *Collaborative Commitment for Action Contributions* and *Collaborative Commitment Inheres in Collaborator* patterns, that belongs a patterns group, for modeling commitments. If the goals are relevant, the *Goal as the Propositional Content of a Collaborative Commitment* pattern can be selected to represent represent the goal that is the propositional content of a collaborative commitment.

Following the selection flow of the patterns, if the conflicts and dependences between goals are relevant for the collaborative commitments, *Goal Conflicts* and *Goal Dependence* pattern can be selected independently of one another. Once the cooperation is modeled and the coordination and/or the communication is out of scope, the pattern application process finishes at the end point EndP1. Otherwise, the coordination and/or communication pattern can be select the same way as when choosing the entry point EP4 and/or EP5.

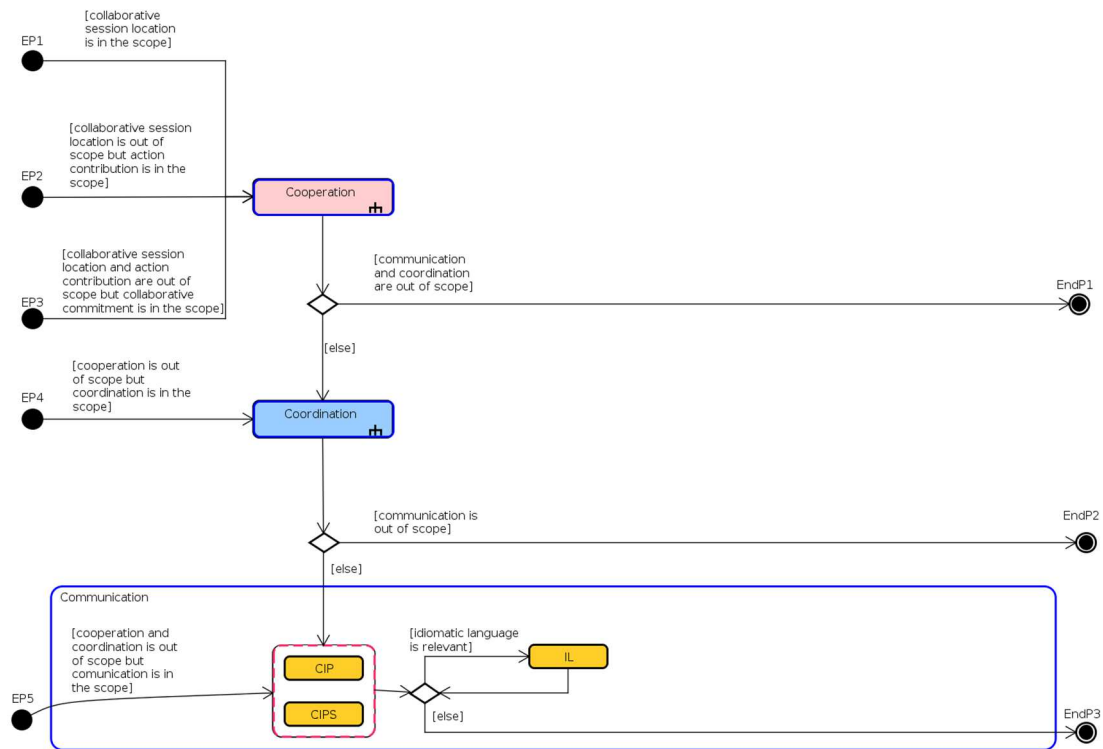


Figura 4 - Detailed View of the Communication Group.

If the entry point EP4 is chosen, the ontology engineer must use first the *Collaborative Session Resource* or *Collaborative Session Resource Simple* patterns that belongs to a variant group for modeling modeling resource participation on the collaborative session. After this, the *Resource Composition* pattern can be chosen if the resource composition is in the scope of the ontology being developed. Then, if the resource nature is in the scope, the *Resource Nature* pattern can be chosen. Once the coordination is modeled, the pattern application process finishes at the end point EndP2, if communication is out of scope. Otherwise, the communication pattern can be select the same way as when choosing the entry point EP5.

When EP5 is chosen, the ontology engineer must use first the *Comminucation Interaction Pattern* or *Communication Interaction Pattern Simple* patterns that belongs to a variant group for modeling modeling communication interactions. Then, if the codification language used in the communication is important, the *Idiomatic Language* pattern is used. Once the communication is modeled, the pattern application process finishes at the end point EndP3.

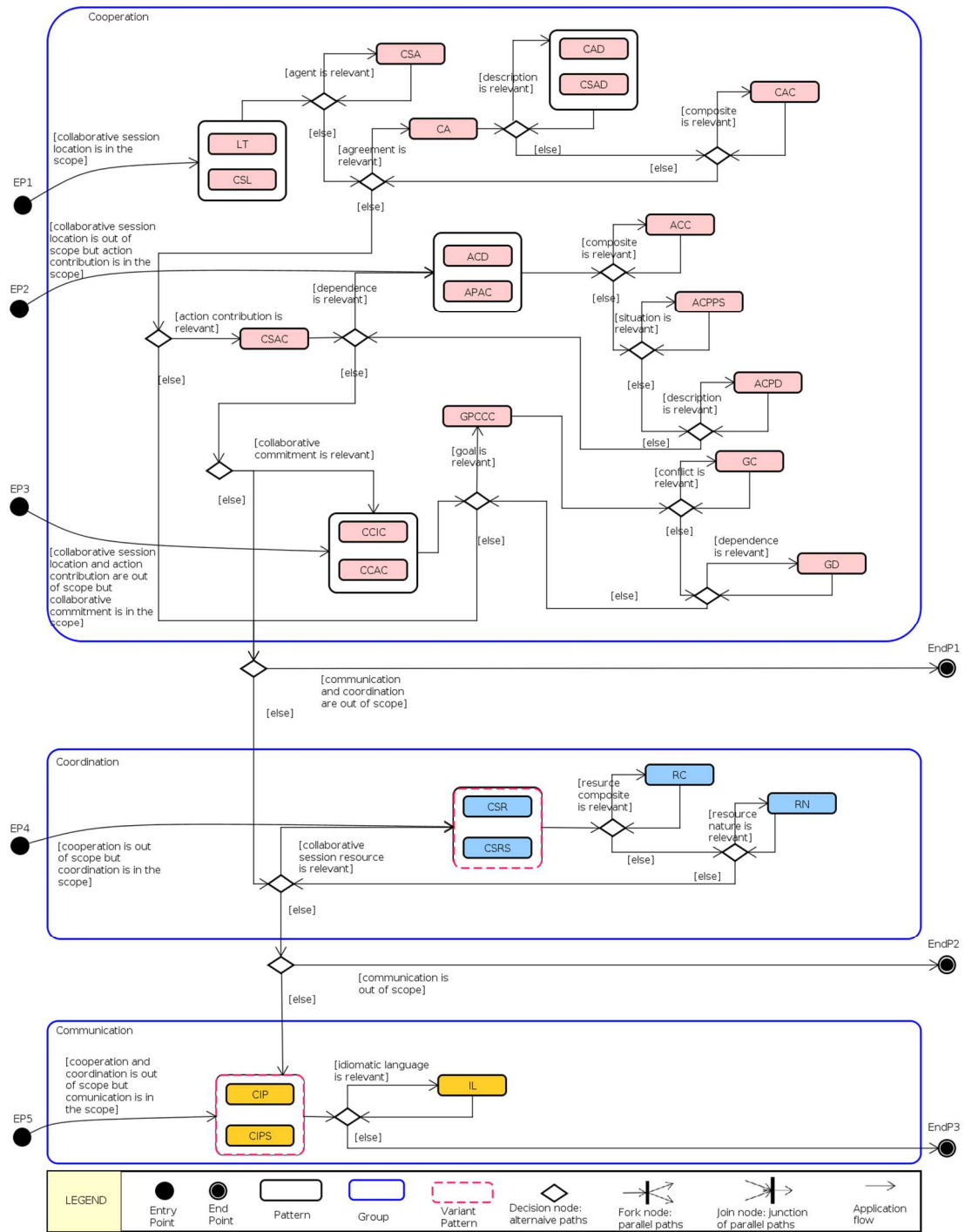


Figure 5 - Detailed View of the C-OPL Process.

4. C-OPL Patterns Descriptions

The description of C-OPL patterns includes the following items:

- **Name:** provides the name of the pattern.
- **Alternative Patterns:** list the alternative patterns for a pattern.
- **Intent:** describes the pattern purpose.
- **Rationale:** describes the rationale underlying the pattern. A short statement answering the following question: What is the pattern rationale?
- **Competency Questions:** describes the competency questions that the pattern aims to answer.
- **Conceptual Model:** depicts the OntoUML diagram representing the pattern elements.

4.1 Cooperation Group

CSL – Collaborative Session Location

Name: Collaborative Session Location (CSL)

Alternative Patterns: --

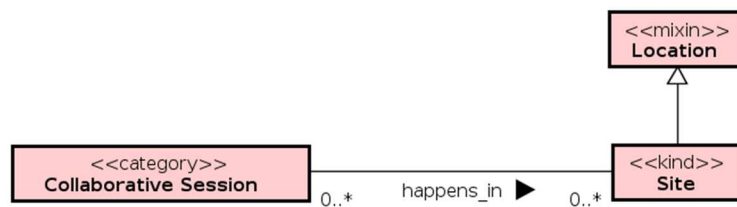
Intent: To represent the site where the collaborative session happens.

Rationale: A Collaborative Session happens in a specific site, which is a location. This pattern makes this site explicit.

Competency Questions:

- Where is the collaborative session?

Conceptual Model:



LT – Location Type

Name: Location Type (LT)

Alternative Patterns: --

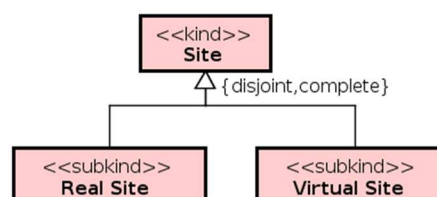
Intent: To represent the type of site where the collaborative session happens.

Rationale: The site where the collaborative session happens can be a real site or a virtual site.

Competency Questions:

- Is the collaborative session site real or virtual?

Conceptual Model:



CSAC – Collaborative Session Action Contributions

Name: Collaborative Session Action Contributions (CSAC)

Alternative Patterns: --

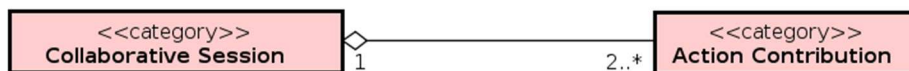
Intent: To represent the action contributions that compose the collaborative session.

Rationale: Two or more action contributions compose a collaborative session.

Competency Questions:

- What are the action contributions that compose the collaborative session?

Conceptual Model:



CSAD – Collaborative Session Agreement Description

Name: Collaborative Session Agreement Description (CSAD)

Alternative Patterns: --

Intent: To represent the descriptions of the collaborative session.

Rationale: A collaborative session has an agreement description, which is a kind of a contract defining the rights and obligations of the roles that may participate in a collaborative session.

Competency Questions:

- What is the agreement description of the collaborative session?

Conceptual Model:



CSA – Collaborative Session Agents

Name: Collaborative Session Agents (CSA)

Alternative Patterns: --

Intent: To represent the agents that participate in the collaborative session.

Rationale: Two or more agents participate in a collaborative session.

Competency Questions:

- What are the agents that participate in the collaborative session?

Conceptual Model:



CA – Collaborative Agreement

Name: Collaborative Agreement (CA)

Alternative Patterns: --

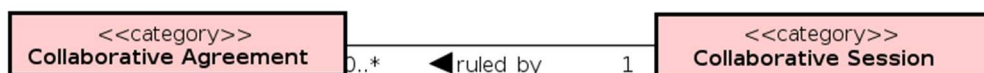
Intent: To represent the collaborative agreement that rules the collaborative session.

Rationale: A collaborative session is ruled by a social relator named collaborative agreement.

Competency Questions:

- What is the collaborative agreement that rules the collaborative session?

Conceptual Model:



CAD – Collaborative Agreement Description

Name: Collaborative Agreement Description (CAD)

Alternative Patterns: --

Intent: To represent the description of the collaborative agreements.

Rationale: A collaborative agreement has a description, which is a kind of a contract defining the rights and obligations of the roles that may participate in a collaborative session.

Competency Questions:

- What is the description of the collaborative agreements?

Conceptual Model:



CAC – Collaborative Agreement Composition

Name: Collaborative Agreement Composition (CAC)

Alternative Patterns: --

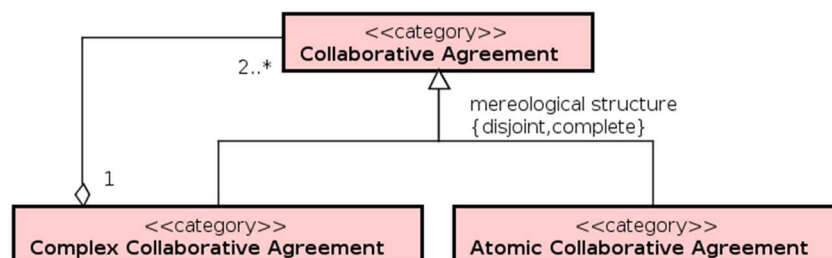
Intent: To represent collaborative agreement mereological decomposition.

Rationale: A collaborative agreement can be composed of other collaborative agreements, either composite collaborative agreement (which can be further decomposed) or simple collaborative agreements (which cannot be decomposed).

Competency Questions:

- Can this collaborative agreement be decomposed?
- What are the components of this collaborative agreement?

Conceptual Model:



CCAC – Collaborative Commitment for Action Contributions

Name: Collaborative Commitment for Action Contributions (CCAC)

Alternative Patterns: --

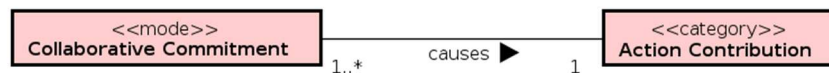
Intent: To represent the collaborative commitment that causes the action contribution.

Rationale: One action contributions is caused by a collaborative commitment.

Competency Questions:

- What is the collaborative commitment that causes the action contribution?

Conceptual Model:



CCIC – Collaborative Commitment Inheres in Collaborator

Name: Collaborative Commitment Inheres in Collaborator (CCIC)

Alternative Patterns: --

Intent: To represent the collaborative commitment that inheres in a collaborator.

Rationale: One collaborative commitment inheres in an collaborator.

Competency Questions:

- What is the collaborator that inheres in a collaborative commitment?

Conceptual Model:



ACC – Acton Contribution Composition

Name: Action Contribution Composition (ACC)

Alternative Patterns: --

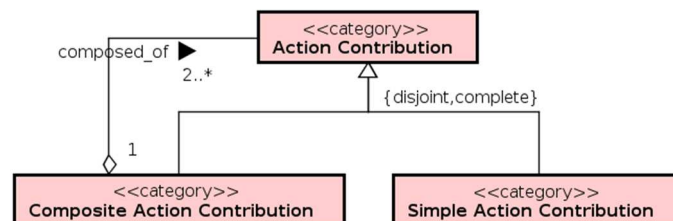
Intent: To represent action contribution mereological decomposition.

Rationale: An action contribution can be composed of other action contributions, either composite action contributions (which can be further decomposed) or simple action contributions (which cannot be decomposed)

Competency Questions:

- Can this action contribution be decomposed?
- What are the components of this action contribution?

Conceptual Model:



ACD – Action Contribution Dependence

Name: Action Contribution Dependence (ACD)

Alternative Patterns: --

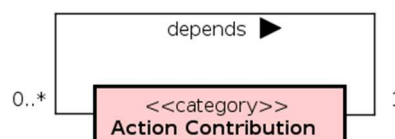
Intent: To represent the dependence between two action contributions

Rationale: An action contribution depends on another goal

Competency Questions:

- Does this action contribution depends on another action contribution?

Conceptual Model:



ACPD – Action Contribution Plan Description

Name: Action Contribution Plan Description (ACPD)

Alternative Patterns: --

Intent: To represent the plan description of the action contributions.

Rationale: A plan description describes the action contributions.

Competency Questions:

- Does this action contribution is described by a plan description?

Conceptual Model:



ACPPS – Action Contribution Pre and Pos Situation

Name: Action Contribution Pre and Pos Situation (ACPPS)

Alternative Patterns: --

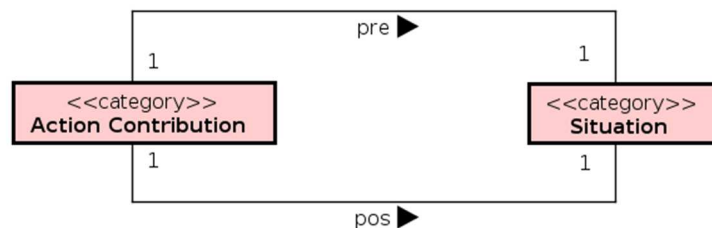
Intent: To represent the pre and pos situation of the action contributions.

Rationale: A pre and pos situation of the action contributions.

Competency Questions:

- What is the situation before the contribution action?
- What is the situation after the contribution action?

Conceptual Model:



APAC - Agent Performs Action Contribution

Name: Agent Performs Action Contribution (APAC)

Alternative Patterns: APCR (Agent Plays Collaborative Role)

Intent: To represent the collaborative roles instantiated by the agents that participate in the collaborative session.

Rationale: The agents participating in a collaborative session instantiate collaborative roles.

Competency Questions:

- What collaborative roles are instantiated by the agents?

Conceptual Model:



GPCCC - Goal as the Propositional Content of a Collaborative Commitment

Name: Goal as the Propositional Content of a Collaborative Commitment (GPCCC)

Alternative Patterns: --

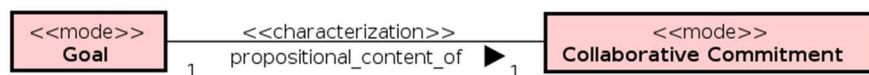
Intent: To represent the goal that is the propositional content of a collaborative commitment

Rationale: A goal that is the propositional content of a collaborative commitment

Competency Questions:

- What is the goal that is the propositional content of a collaborative commitment?

Conceptual Model:



GC – Goal Conflict

Name: Goal Conflict (GC)

Alternative Patterns: --

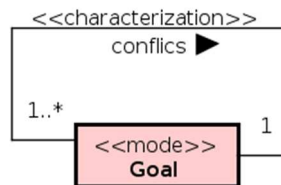
Intent: To represent the conflict between two goals

Rationale: A goal conflicts with another goal

Competency Questions:

- Does this goal conflict with another goal?

Conceptual Model:



GD – Goal Dependence

Name: Goal Dependence (GD)

Alternative Patterns: --

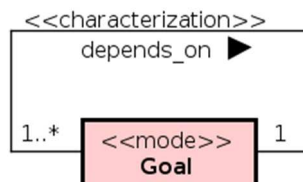
Intent: To represent the dependence between two goals

Rationale: A goal dependence from another goal

Competency Questions:

- Does this goal depends on another goal?

Conceptual Model:



4.2 Coordination Group

CSRS - Collaborative Session Resource Simple

Name: Collaborative Session Resources Simple (CSRS)

Alternative Patterns: CSR

Intent: To represent the resources that participate in the collaborative session.

Rationale: One or more resources participate in a collaborative session.

Competency Questions:

- What are the resources that participate in the collaborative session?

Conceptual Model:



CSR - Collaborative Session Resource

Name: Collaborative Session Resource (CSR)

Alternative Patterns: CSRS (Collaborative Session Resource Simple)

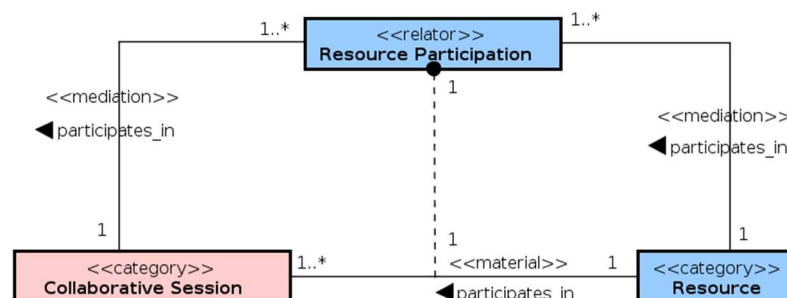
Intent: To represent the participation of resources in collaborative session (with the mediation of a resource participation relator).

Rationale: A collaborative session has one or more resource participation, which refers to one resource.

Competency Questions:

- What are the resources that participate on the collaborative session?

Conceptual Model:



RC - Resource Composition

Name: Resource Composition (RC)

Alternative Patterns: --

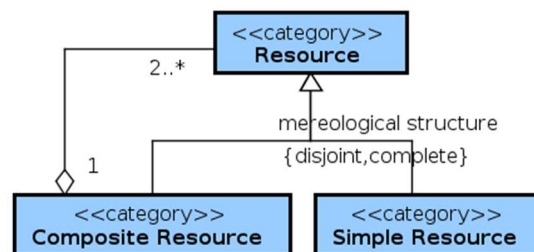
Intent: To represent resource mereological decomposition.

Rationale: A resource can be composed of other resources, either composed resources (which can be further decomposed) or simple resources (which cannot be decomposed)

Competency Questions:

- Can this resource be decomposed?
- What are the components of this resource?

Conceptual Model:



RN - Resource Nature

Name: Resource Nature (RN)

Alternative Patterns: --

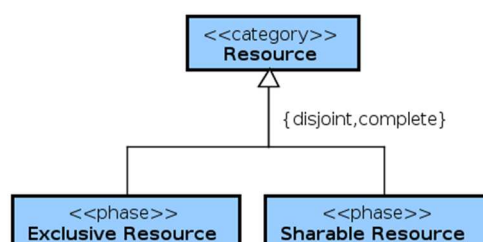
Intent: To represent resource types in terms of shareability.

Rationale: A resource can be either sharable or exclusive

Competency Questions:

- Can this resource be shared?

Conceptual Model:



4.3 Communication Group

CIPS – Communication Interaction Pattern Simple

Name: Communication Interaction Pattern Simple (CIPS)

Alternative Patterns: Communication Interaction Pattern (CIP)

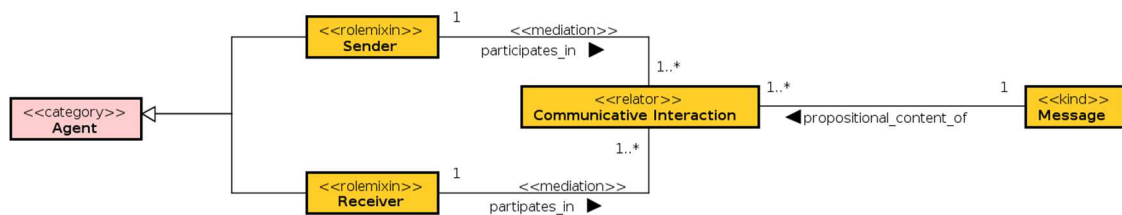
Intent: To represent an interaction pattern connecting a sender and a receiver

Rationale: A communication interaction consists in a sender sending a message to a receiver.

Competency Questions:

- Who is the sender in the communicative interaction?
- Who is the receiver in the communicative interaction?
- What is the message of the communicative interaction?

Conceptual Model



CIP – Communication Interaction Pattern

Name: Communication Interaction Pattern

Alternative Patterns: Communication Interaction Pattern Simple (CIPS)

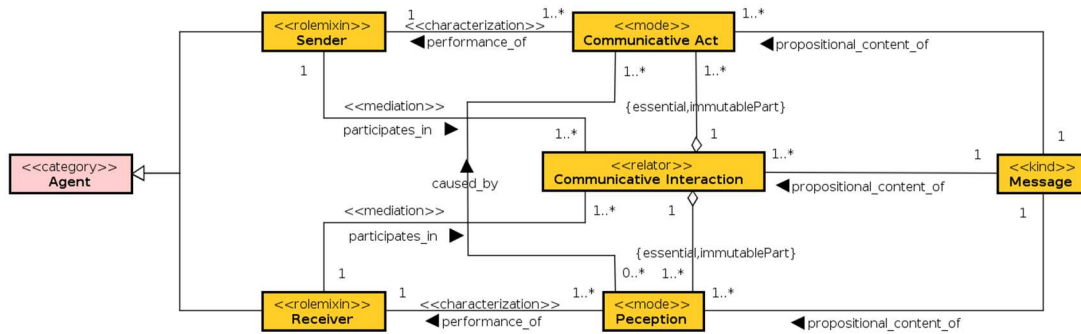
Intent: To represent in detail an interaction pattern connecting a sender and a receiver

Rationale: A communication interaction consists in a communicative act performed by the sender and a perception performed by the receiver. The message is the propositional content of the communicative act and the perception.

Competency Questions:

- Who is the sender in the communicative interaction?
- Who is the receiver in the communicative interaction?
- What is the message of the communicative interaction?

Conceptual Model:



IL - Idiomatic Language

Name: Idiomatic Language (IL)

Alternative Patterns: --

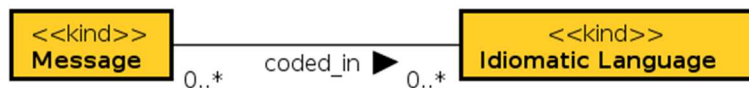
Intent: To represent the idiomatic language codifying the message.

Rationale: A message may be codified in an idiomatic language.

Competency Questions:

- What is the idiomatic language that codifies the message?

Conceptual Model:



5. References

1. Falbo, R. A., Barcellos, M.P., Nardi, J.C., and Guizzardi, G., *Organizing Ontology Design Patterns as Ontology Pattern Languages*. Proceedings of the 10th Extended Semantic Web Conference - ESWC 2013, 2013
2. Oliveira, F. F. (2009). *Uma ontologia de colaboração e suas aplicações* (Master's thesis, Universidade Federal do Espírito Santo)
3. Ellis, C. A., Gibbs, S. J., & Rein, G. (1991). *Groupware: some issues and experiences*. Communications of the ACM, 34(1), 39-58.
4. Oliveira, F. F., Antunes, J. C., & Guizzardi, R. S. (2007, October). *Towards a collaboration ontology*. In Proc. of the Snd Brazilian Workshop on Ontologies and Metamodels for Software and Data Engineering. João Pessoa.
5. Oleary, D. E. *Using AI in knowledge management: Knowloge bases and ontologies*. IEEE-IS, MAY/JUNE 1998.

APÊNDICE B – Pesquisa de Avaliação do Suporte Automatizado e da Abordagem de Catálogo de DROPs

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

O questionário a seguir visa avaliar a proposta de uma abordagem metodológica para uso de catálogos de DROPs (Domain-related Ontology Pattern) no desenvolvimento de ontologias. A sistematização da abordagem de catálogos de DROPs para a construção de ontologias de domínio específico tem por objetivo prover ganhos ao processo de modelagem de ontologias de domínio. Para tanto, foi criado um módulo para prover suporte automatizado a abordagem citada por meio de ferramenta computacional (OLED) para a criação e reutilização de catálogos de DROPs no processo de modelagem de ontologias, fornecendo, assim, sistematicidade para os catálogos de DROPs e proporcionando um nível de abstração maior à modelagem por meio da utilização dos DROPs dos catálogos.

As informações coletadas por este questionário serão compiladas e avaliadas para mensuração da contribuição que a abordagem e o suporte computacional trouxeram ao processo de construção de ontologias de domínio.

As perguntas são objetivas e serão respondidas muito rapidamente. Desde já agradecemos a sua participação de contribuição com essa pesquisa.

A versão do OLED com a implementação da abordagem de catálogos de DROPs pode ser acessada pelo link:

https://www.dropbox.com/s/qe23luknfwjwfne/oled_2.2.0.jar?dl=0

A documentação demonstrando os recursos presentes na versão do OLED com a implementação da abordagem de catálogos de DROPs pode ser acessada pelo link:

https://www.dropbox.com/s/hcaqlap92aq187c/suporte_automatizado_catalogos_drops_oled.pdf?dl=0

PRÓXIMA



Página 1 de 21

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Modelagem de ontologias

1. Quantos anos de experiência você tem ou teve com modelagem de ontologias? *

- Até 1 ano
- Até 3 anos
- Até 5 anos
- Até 7 anos
- Mais de 7 anos

VOLTAR

PRÓXIMA

Página 2 de 21

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Modelagem de ontologias

2. Como você enxerga a modelagem de ontologias hoje em dia?

*

- Complexa
- Improdutiva
- Vaga
- Simples
- Produtiva
- Objetiva

VOLTAR

PRÓXIMA

Página 3 de 21

Nunca envie senhas pelo Formulários Google.

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Modelagem de ontologias

3. O que você, normalmente, utiliza para a modelagem de ontologias? *

- Abordagens, metodologias, etc.
- Ferramenta computacional
- Linguagens de modelagem ontológicas (OntoUML, ...)
- Padrões ontológicos (DROPs, FOPs, ...)
- Catálogos de padrões ontológicos
- OPLs
- Nenhuma das opções anteriores

VOLTAR

PRÓXIMA

Página 4 de 21

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Modelagem de ontologias

4. Com que frequência você utiliza ferramenta computacional para modelagem de ontologias? *

- Sempre
- Maioria das vezes
- De vez em quando
- Raramente
- Nunca

VOLTAR

PRÓXIMA

 Página 5 de 21

Nunca envie senhas pelo Formulários Google.

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

Modelagem de ontologias

5. Quando você utiliza ferramenta computacional para a modelagem de ontologias, essas ferramentas suportam?

- Abordagens, metodologias, etc.
- Linguagens de modelagem ontológicas (OntoUML, ...)
- Padrões ontológicos (DROPs, FOPs, ...)
- Catálogos de padrões ontológicos
- OPLs
- Nenhuma das opções anteriores
- Não utilizo ferramenta computacional

VOLTAR

PRÓXIMA

Página 6 de 21

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Modelagem de ontologias

6. Qual a sua avaliação da utilização de ferramenta computacional para apoiar a modelagem de ontologias? *

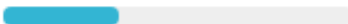
- Muito útil
- Útil
- Indiferente
- Inútil
- Muito inútil

Justificativa: *

Sua resposta

VOLTAR

PRÓXIMA

 Página 7 de 21

Nunca envie senhas pelo Formulários Google.

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Abordagem de catálogos de DROPs

7. Qual a sua avaliação do suporte automatizado implementado no OLEP e da abordagem metodológica proposta para criação de catálogos de DROPs? *

- Muito útil
- Útil
- Indiferente
- Inútil
- Muito inútil

Justificativa: *

Sua resposta

VOLTAR

PRÓXIMA

Página 8 de 21

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Abordagem de catálogos de DROPs

8. Em sua avaliação, o suporte automatizado implementado no OLEP para criação de catálogos de DROPs e a abordagem metodológica proposta favorecem a produtividade na criação desses catálogos? *

- Favorece muito
- Favorece pouco
- Indiferente
- Desfavorece pouco
- Desfavorece muito

Justificativa: *

Sua resposta

VOLTAR

PRÓXIMA

 Página 9 de 21

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Abordagem de catálogos de DROPs

9. Em sua avaliação, o suporte automatizado implementado no OLEP para criação de catálogos de DROPs e a abordagem metodológica proposta favorecem a qualidade desses catálogos e dos DROPs deles? *

- Favorece muito
- Favorece pouco
- Indiferente
- Desfavorece pouco
- Desfavorece muito

Justificativa: *

Sua resposta

VOLTAR

PRÓXIMA

Página 10 de 21

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Abordagem de catálogos de DROPs

10. Em sua avaliação, o suporte automatizado implementado no OLEP para criação de catálogos de DROPs e a abordagem metodológica proposta favorecem o compartilhamento desses catálogos e dos DROPs deles? *

- Favorece muito
- Favorece pouco
- Indiferente
- Desfavorece pouco
- Desfavorece muito

Justificativa: *

Sua resposta

VOLTAR

PRÓXIMA

 Página 11 de 21

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Abordagem de catálogos de DROPs

11. Em sua avaliação, o suporte automatizado implementado no OLE para criação de catálogos de DROPs e a abordagem metodológica proposta favorecem a configuração desses catálogos e dos DROPs deles? *

- Favorece muito
- Favorece pouco
- Indiferente
- Desfavorece pouco
- Desfavorece muito

Justificativa: *

Sua resposta

VOLTAR

PRÓXIMA

Página 12 de 21

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Abordagem de catálogos de DROPs

12. Qual a sua avaliação do suporte automatizado implementado no OLEP para reutilização de catálogos de DROPs? *

- Muito útil
- Útil
- Indiferente
- Inútil
- Muito inútil

Justificativa: *

Sua resposta

VOLTAR

PRÓXIMA

Página 13 de 21

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Abordagem de catálogos de DROPs

13. Em sua avaliação, o suporte automatizado implementado no OLEP para reutilização de catálogos de DROPs favorece produtividade na modelagem de ontologias? *

- Favorece muito
- Favorece pouco
- Indiferente
- Desfavorece pouco
- Desfavorece muito

Justificativa: *

Sua resposta

VOLTAR

PRÓXIMA

Página 14 de 21

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Abordagem de catálogos de DROPs

14. Em sua avaliação, o suporte automatizado implementado no OLEP para reutilização de catálogos de DROPs favorece a qualidade das ontologias criadas com eles? *

- Favorece muito
- Favorece pouco
- Indiferente
- Desfavorece pouco
- Desfavorece muito

Justificativa: *

Sua resposta

VOLTAR

PRÓXIMA

Página 15 de 21

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Abordagem de catálogos de DROPs

15. Em sua avaliação, o suporte automatizado implementado no OLEP para reutilização de catálogos de DROPs favorece o compartilhamento de ontologias construídas com esses catálogos? *

- Favorece muito
- Favorece pouco
- Indiferente
- Desfavorece pouco
- Desfavorece muito

Justificativa: *

Sua resposta

VOLTAR

PRÓXIMA

 Página 16 de 21

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Abordagem de catálogos de DROPs

16. Em sua avaliação, a abordagem metodológica proposta para a criação de catálogos de DROPs e o suporte automatizado implementados no OLED adicionam ao processo de construção de catálogo de DROPs mais? *

- Produtividade
- Praticidade
- Robustez
- Sistematicidade
- Qualidade
- Improdutividade
- Complexidade
- Fraqueza
- Desorganização
- Deficiência

Justificativa: *

Sua resposta

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Abordagem de catálogos de DROPs

17. Em sua avaliação, a abordagem metodológica proposta para catálogos de DROPs e o suporte automatizado implementado no OLEP adicionam ao processo de modelagem de ontologias mais? *

- Produtividade
- Praticidade
- Robustez
- Sistemática
- Qualidade
- Improdutividade
- Complexidade
- Fraqueza
- Desorganização
- Deficiência

Justificativa: *

Sua resposta

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Abordagem de catálogos de DROPs

18. Você já havia utilizado a abordagem de catálogos de DROPs com suporte automatizado para modelagem de ontologias? *

- Sim
- Não

VOLTAR

PRÓXIMA

 Página 19 de 21

Nunca envie senhas pelo Formulários Google.

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Abordagem de catálogos de DROPs

19. Em sua avaliação, no geral, você utilizaria a abordagem de catálogos de DROPs com suporte automatizado implementado para a modelagem de ontologias? *

- Sim
- Indiferente
- Não

Justificativa: *

Sua resposta

VOLTAR

PRÓXIMA

 Página 20 de 21

Nunca envie senhas pelo Formulários Google.

Avaliação de suporte automatizado para desenvolvimento de ontologias utilizando catálogos de DROPs

*Obrigatório

Abordagem de catálogos de DROPs

20. Em sua avaliação, no geral, o suporte automatizado e a abordagem de catálogos de DROPs contribuem com o processo de modelagem de ontologias? *

- Sim
- Indiferente
- Não

Justificativa: *

Sua resposta

VOLTAR

ENVIAR

 Página 21 de 21

Nunca envie senhas pelo Formulários Google.
