

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

Glaice Kelly da Silva Quirino

Uma Notação Visual para Representação de Linguagens de Padrões Ontológicos

VITÓRIA
2016

Glaice Kelly da Silva Quirino

Uma Notação Visual para Representação de Linguagens de Padrões Ontológicos

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.

Orientador(a): Monalessa Perini Barcellos
Coorientador: Ricardo de Almeida Falbo

VITÓRIA
2016

Glaice Kelly da Silva Quirino

Uma Notação Visual para Representação de Linguagens de Padrões Ontológicos

COMISSÃO EXAMINADORA

Prof. Monalessa Perini Barcellos, D. Sc.
Universidade Federal do Espírito Santo (UFES)
Orientador

Prof. Ricardo de Almeida Falbo, D. Sc.
Universidade Federal do Espírito Santo (UFES)
Coorientador

Prof^a. Renata Silva Souza Guizzardi, Ph.D.
Universidade Federal do Espírito Santo (UFES)

Prof^a. Maria Luiza Machado Campos, D. Sc.
Universidade Federal do Rio de Janeiro (UFRJ)

Vitória, 31 de março de 2016

Dedico esta dissertação à minha família e ao meu esposo, Warlen, que sempre me apoiaram, incentivaram e vivenciaram cada segundo dessa trajetória ao meu lado.

AGRADECIMENTOS

Primeiramente, agradeço a Deus por me abençoar, fortalecer e guiar em toda essa jornada.

Agradeço aos meus pais, João e Luzia, e ao meu irmão, Wancharle, por todo amor e carinho e por todo apoio e incentivo dado para que eu pudesse realizar meus sonhos.

Agradeço ao Warlen, meu amado esposo e companheiro de tantas aventuras da vida, por todo amor, compreensão e apoio dado diariamente.

Agradeço aos meus professores orientadores, Monalessa e Falbo, por toda orientação e dedicação. Vocês foram fundamentais na realização desse trabalho. Obrigada por serem tão pacientes e por me incentivarem a melhorar cada vez mais.

Agradeço aos membros da banca de defesa do mestrado, as professoras Renata e Maria Luiza, por cederem seu tempo participando da defesa e contribuindo para a melhoria deste trabalho.

Agradeço ao Julio Nardi, Fabiano Ruy e Maria das Graças Teixeira por compartilharem comigo experiências e trabalhos acadêmicos que tanto contribuíram para a concretização deste trabalho.

Agradeço aos colegas e professores do NEMO, por me ajudarem em tantos momentos e por todo apoio e incentivo.

Agradeço a todos meus familiares e amigos, principalmente os amigos do GASPE, que sempre dedicaram muita atenção, carinho e apoio.

Por fim, agradeço a todos que contribuíram de alguma forma com este trabalho.

RESUMO

Reúso tem sido apontado como uma abordagem promissora para a Engenharia de Ontologias. A reutilização em ontologias permite acelerar o processo de desenvolvimento, além de melhorar a qualidade das ontologias resultantes, uma vez que promove a aplicação de boas práticas. Para favorecer o reúso, o uso de padrões (*patterns*) tem sido explorado na Engenharia de Ontologias. Um padrão pode ser definido como uma solução bem-sucedida para um problema recorrente. Assim, padrões ontológicos (PO) apresentam soluções para problemas de modelagem recorrentes. Padrões podem ser organizados em uma Linguagem de Padrões (LP), que representa os padrões e suas relações em uma rede e define um processo para seleção e utilização dos padrões para resolução sistemática de problemas. Nesse sentido, uma Linguagem de Padrões Ontológicos (LPO) fornece, além dos padrões, diretrizes para sua seleção, incluindo a sequência em que podem ser usados, as variações existentes e os caminhos possíveis, entre outros. Para facilitar o uso de uma LPO, o processo e as relações entre os padrões devem ser representados de forma clara, não ambígua e completa. Notações visuais podem ser utilizadas para prover uma representação visual da LPO e são um importante meio de comunicação entre *stakeholders*, uma vez que se acredita que elas transmitam informações de forma mais eficaz do que texto. Para facilitar o entendimento e potencializar o uso de uma LPO, sua notação visual deve ser cognitivamente rica. Este trabalho propõe uma notação visual para representação de LPOs. Como bases para a proposta, foram utilizados os resultados de um mapeamento sistemático que investigou notações visuais em Linguagens de Padrões de Software e resultados de um estudo experimental em que se avaliou a notação utilizada para representar uma LPO. Buscando-se obter uma notação cognitivamente rica, os princípios de PoN (*Physics of Notation*) foram considerados no desenvolvimento da notação. Como prova de conceito, a notação visual proposta foi utilizada na reengenharia de uma LPO. Como avaliação preliminar da proposta, foi realizado um estudo experimental.

Palavras-chave: Linguagem de Padrões Ontológicos, Ontologia, Notação Visual.

ABSTRACT

Reuse has been pointed out as a promising approach for Ontology Engineering. Reuse in ontologies allows speeding up the development process and improves the quality of the resulting ontologies, since it promotes the application of good practices. The use of patterns as an approach to encourage reuse has been explored in Ontology Engineering. A pattern can be defined as a successful solution for a recurring problem. Thus, Ontology Patterns (OPs) address solutions for recurring modeling problems. Patterns can be arranged in a Pattern Language (PL), which represents the patterns and their relationships in a network and defines a process for selection and use of patterns for systematic problem solving. In this sense, Ontology Pattern Languages (OPLs) provide, in addition to the patterns, guidelines for their selection, including the sequence they can be applied, variations and possible paths to be followed, among others. In order to make easier using an OPL, the process and relationships between the patterns should be represented in a clear, unambiguous and complete way. Visual Notations can be used to provide a visual representation of the OPL. They are an important means of communication between stakeholders, since it is believed that they transmit information more effectively than text. To facilitate understanding an OPL and strengthen its use, its visual notation must be cognitively rich. This work proposes a visual notation for representing OPLs. As basis for the proposal, we used the results of a systematic mapping that investigated visual notations for Software Pattern Languages. In addition, we used results of an experimental study that evaluated the notation used to represent an OPL. Aiming to obtain a cognitively rich notation, the principles of PoN (Physics of Notation) were considered during the notation development. As proof of concept, we used the proposed visual notation to reengineer an OPL. Finally, as a preliminary evaluation of the proposal, we performed an experimental study.

Keywords: Ontology Pattern Language, Ontology, Visual Notation.

SUMÁRIO

INTRODUÇÃO	10
1.1 CONTEXTO.....	10
1.2 MOTIVAÇÃO.....	12
1.3 OBJETIVOS DA PESQUISA	13
1.4 MÉTODO DE PESQUISA.....	13
1.5 ORGANIZAÇÃO DA DISSERTAÇÃO.....	15
CAPÍTULO 6 LINGUAGENS DE PADRÕES ONTOLÓGICOS E NOTAÇÕES VISUAIS.....	17
2.1 ONTOLOGIAS.....	17
2.2 PADRÕES ONTOLÓGICOS	18
2.3 LINGUAGENS DE PADRÕES ONTOLÓGICOS	20
2.3.1. <i>Um Exemplo de Linguagem de Padrões Ontológicos</i>	21
2.4 NOTAÇÕES VISUAIS.....	25
2.4.1. <i>PoN-Systematized (PoN-S)</i>	28
2.5 CONSIDERAÇÕES FINAIS	33
CAPÍTULO 3 NOTAÇÕES VISUAIS EM LINGUAGENS DE PADRÕES DE SOFTWARE: MAPEAMENTO SISTEMÁTICO.....	34
3.1 LINGUAGENS DE PADRÕES	34
3.2 MÉTODO ADOTADO	35
3.3 PROTOCOLO DE PESQUISA	36
3.4 EXECUÇÃO DO MAPEAMENTO E SÍNTESE DOS DADOS	38
3.5 DISCUSSÕES	54
3.6 CONSIDERAÇÕES FINAIS	57
CAPÍTULO 4 ESTUDO EXPERIMENTAL PARA AVALIAÇÃO DE UMA LINGUAGEM DE PADRÕES ONTOLÓGICOS.....	59
4.1 ISO-BASED SOFTWARE PROCESS ONTOLOGY PATTERN LANGUAGE (ISP – OPL)	59
4.2 PLANEJAMENTO DO ESTUDO.....	61
4.3 RESULTADOS.....	64
4.3.1 <i>Percepção acerca da LPO</i>	64
4.3.2 <i>Percepção acerca do uso de LPOs no desenvolvimento de ontologias de domínio</i>	66
4.3.3 <i>Entrevistas</i>	67
4.4 DISCUSSÕES	68
4.5 AMEAÇAS À VALIDADE.....	69
4.6 CONSIDERAÇÕES FINAIS	70
CAPÍTULO 5 EM DIREÇÃO A UMA NOTAÇÃO VISUAL PARA LPOS: PRIMEIROS PASSOS.....	71

5.1	NOTAÇÃO VISUAL PARA LPOS: UMA VERSÃO PRELIMINAR.....	71
5.2	SERVICE ONTOLOGY PATTERN LANGUAGE (S-OPL).....	72
5.2.1	<i>Padrões Ontológicos de S-OPL</i>	73
5.2.2	<i>Modelo de Processo de S-OPL</i>	78
5.3	ESTUDO EXPERIMENTAL PARA AVALIAÇÃO DE S-OPL E DA NOTAÇÃO VISUAL ADOTADA.....	81
5.3.1	<i>Resultados</i>	83
5.3.1.1	<i>Percepção acerca da LPO</i>	83
5.3.1.2	<i>Percepção acerca do Uso de LPOs no Desenvolvimento de Ontologias de Domínio</i>	85
5.3.2	<i>Entrevistas</i>	87
5.3.3	<i>Discussões</i>	88
5.3.4	<i>Ameaças à Validade</i>	89
5.4	CONSIDERAÇÕES FINAIS.....	89
CAPÍTULO 6 OPL-ML: NOTAÇÃO VISUAL PARA REPRESENTAÇÃO DE LINGUAGENS DE PADRÕES ONTOLÓGICOS		91
6.1	INTRODUÇÃO	91
6.2	SINTAXE ABSTRATA	92
6.3	SINTAXE CONCRETA.....	97
6.4	APLICAÇÃO DOS PRINCÍPIOS DE PON NO DESIGN DA SINTAXE CONCRETA.....	100
6.5	REENGENHARIA DA REPRESENTAÇÃO VISUAL DE UMA LPO	106
6.6	AVALIAÇÃO DE OPL-ML.....	113
6.6.1	<i>Resultados do Survey</i>	115
6.6.1.1	<i>Percepção acerca do entendimento do mecanismo de funcionamento geral de LPOs</i>	115
6.6.1.2	<i>Percepção acerca das Mudanças realizadas na LPO</i>	116
6.6.2	<i>Discussões</i>	117
6.6.3	<i>Ameaças à Validade</i>	118
6.7	CONSIDERAÇÕES FINAIS	118
CAPÍTULO 7 CONCLUSÃO		120
7.1	CONSIDERAÇÕES FINAIS.....	120
7.2	CONTRIBUIÇÕES	123
7.3	TRABALHOS FUTUROS.....	123
REFERÊNCIAS BIBLIOGRÁFICAS		125
APÊNDICE A		134
A1.	TERMO DE CONSENTIMENTO UTILIZADO NOS ESTUDOS EXPERIMENTAIS	134
A2.	FORMULÁRIO DE CARACTERIZAÇÃO UTILIZADO NOS ESTUDOS EXPERIMENTAIS	135
A3.	QUESTIONÁRIO UTILIZADO NOS ESTUDOS EXPERIMENTAIS	138
A4.	QUESTIONÁRIO USADO NAS ENTREVISTAS DOS ESTUDOS EXPERIMENTAIS:.....	143
A5.	QUESTIONÁRIO UTILIZADO NO SURVEY.	145

Introdução

Este capítulo apresenta o contexto, a motivação e os objetivos do trabalho, bem como o método de pesquisa adotado e a organização do texto desta dissertação.

1.1 Contexto

No âmbito da Computação, em áreas como Inteligência Artificial, Engenharia de Software e Web Semântica, uma ontologia é um artefato que descreve certa realidade, segundo algum propósito. Como qualquer artefato, ontologias têm um ciclo de vida: elas são projetadas, implementadas, avaliadas, alteradas, reutilizadas etc. (GANGEMI; PRESUTTI, 2009).

O desenvolvimento de ontologias não é uma tarefa trivial e mesmo especialistas enfrentam dificuldades. Embora a Engenharia de Ontologias proveja diversas ferramentas para auxiliar no desenvolvimento de ontologias, muitas dificuldades ainda persistem. Uma abordagem que pode contribuir para o desenvolvimento de ontologias é a reutilização, uma vez que os fragmentos de ontologias previamente desenvolvidas podem ser reutilizados na construção de novas (FALBO *et al.*, 2013). A adoção de reuso na engenharia de ontologias permite acelerar o processo de desenvolvimento, economizando tempo e custos, além de melhorar a qualidade das ontologias resultantes, uma vez que promove a aplicação de boas práticas (POVEDA-VILLALÓN *et al.*, 2010).

Contudo, reuso é ainda uma das áreas mais desafiadoras e negligenciadas na Engenharia de Ontologias. Algumas possíveis razões para isso são: o tamanho e a complexidade das ontologias existentes e potenciais candidatas ao reuso, a obscuridade do *design rationale* na maioria das ontologias e a fragilidade das ferramentas de apoio ao engenheiro de ontologias (GANGEMI; PRESUTTI, 2009).

Algumas ontologias pequenas e simples como FOAF (*Friend Of A Friend*) (BRICKLEY; MILLER, 2005) e SKOS (*Simple Knowledge Organisation for the Web*) (MILES; BRICKLEY, 2005) demonstraram o potencial de ontologias reutilizáveis e levaram à percepção de que ontologias podem ser construídas a partir de fragmentos de ontologia que tratam problemas específicos. Esses fragmentos funcionam como blocos de construção que podem ser combinados para tratar os problemas relevantes para a ontologia a ser desenvolvida e são chamados padrões ontológicos. Um Padrão Ontológico (PO) descreve

um problema de modelagem recorrente que emerge a partir do desenvolvimento de ontologias em contextos específicos e apresenta uma solução para ele (FALBO *et al.*, 2013b)

Típicamente, POs são disponibilizados em catálogos, porém essa abordagem não explora totalmente o potencial de reúso, uma vez que em um catálogo convencional perde-se o senso de conexão entre os POs. Diferente de catálogos de padrões, uma Linguagem de Padrões Ontológicos (LPO) favorece ainda mais o reúso ao prover uma rede de padrões ontológicos interconectados que fornece suporte holístico para o desenvolvimento de ontologias em um dado campo. Além da rede de padrões, uma LPO provê um processo que guia a ordem de aplicação dos padrões e sugere quais padrões utilizar de acordo com os problemas a serem tratados (FALBO *et al.*, 2016).

O termo linguagem de padrões é reutilizado da Engenharia de Software, área na qual padrões têm sido estudados e aplicados há décadas. Em Engenharia de Software uma linguagem de padrões é uma rede de padrões inter-relacionados que define um processo para sistematicamente resolver problemas relacionados ao desenvolvimento de software (DEUTSCH, 2004 e BUSCHMANN *et al.*, 2007). Assim, vale ressaltar que, embora o termo linguagem seja utilizado, ele não tem seu significado literal, uma vez que em uma linguagem de padrões tipicamente não se define uma linguagem de fato (FALBO *et al.*, 2013).

Segundo Moody (2009), notações visuais desempenham papel importante na comunicação, permitindo a representação de abstrações que contribuem para a visualização de informações que podem levar a um entendimento mais eficaz do que o obtido a partir da leitura de textos. Se o termo notação visual for considerado em seu sentido literal, toda e qualquer notação passível de ser vista poderia ser considerada uma notação visual, até mesmo notações puramente textuais. No entanto, no contexto deste trabalho, usa-se o termo notação visual em linha com o que define Moody (2009), que afirma que uma notação visual (também chamada linguagem visual, notação gráfica ou notação diagramática) consiste em um conjunto de símbolos gráficos, um conjunto de regras de composição e definições do significado de cada símbolo, os quais são utilizados para a elaboração de diagramas. Nesse sentido, LPOs devem ser representadas visualmente por meio de diagramas que representem os padrões propriamente ditos, as relações entre eles e o processo que guia sua aplicação.

Linguagem de padrões é um tema consideravelmente recente na Engenharia de Ontologias. Com isso, aspectos relacionados à sua representação visual ainda não têm sido foco de investigação de outros trabalhos. Uma vez que na Engenharia de Software o tema é relativamente maduro, conhecimento acerca de representação visual de linguagens de padrões na Engenharia de Software pode ser útil para a representação visual de LPOs.

1.2 Motivação

Notações visuais são amplamente usadas em Engenharia de Software e têm estado presente na pesquisa e na prática desde seu início. A importância das notações visuais é reconhecida devido à sua eficácia na transmissão de informações (MOODY *et al.*, 2010). Segundo Ware (2004), pessoas adquirem mais informações através da visão do que através de todos os outros sentidos combinados. Desse modo, a representação visual de dados propicia o fornecimento de informações que podem ser rapidamente interpretadas (WARE, 2004). Além disso, os diagramas podem transmitir informações de forma mais concisa e precisa do que a linguagem natural (MOODY, 2009).

Os benefícios providos pelo uso de notações visuais também podem ser percebidos no âmbito da Engenharia de Ontologias. Ontologias são, tipicamente, representadas por meio de diagramas, o que facilita, por exemplo, a identificação dos conceitos tratados na ontologia e das relações entre eles e contribui para a percepção e o entendimento da ontologia como um todo.

Considerando-se que, como discutido na seção anterior, LPOs podem ser utilizadas para apoiar o desenvolvimento de ontologias a fim de diminuir a complexidade dessa tarefa, uma representação gráfica dos POs, suas relações e a sequência em que podem ser aplicados torna-se crucial para o efetivo entendimento e uso de LPOs. Vale destacar que a representação visual deve ser clara, não ambígua e completa.

No que se refere à representação do processo para guiar a aplicação dos padrões, é necessário que ela forneça diretrizes para a utilização dos padrões, incluindo a sequência em que eles podem ser usados, as variações existentes, os caminhos possíveis, entre outras. (FALBO *et al.*, 2013).

A adoção de LPOs como uma abordagem para reutilização na Engenharia de Ontologias foi originalmente proposta por pesquisadores do Núcleo de Estudos em Modelagem Conceitual e Ontologias (NEMO) (FALBO *et al.*, 2013), grupo de pesquisa no qual este trabalho foi desenvolvido. Nos últimos anos foram desenvolvidas no NEMO LPOs para os domínios processos de software (FALBO *et al.*, 2013) (RUY *et al.*, 2015a), organizações (FALBO *et al.*, 2014), medição (BARCELLOS *et al.*, 2014) e serviços (FALBO *et al.*, 2016). A representação visual do processo dessas LPOs foi feita utilizando-se variações do diagrama de atividades da UML. No entanto, devido à inexistência de uma notação bem definida capaz de tratar as necessidades de representação de LPOs, foram identificados problemas, limitações e inconsistências nas representações do processo das LPOs desenvolvidas.

Dessa forma, a importância da representação visual para o efetivo uso de LPOs e a ausência de uma notação proposta com essa finalidade revelam uma oportunidade de pesquisa que foi explorada neste trabalho.

1.3 Objetivos da Pesquisa

Este trabalho tem como objetivo geral **definir uma notação para representação visual de linguagens de padrões ontológicos**. Esse objetivo geral pode ser detalhado nos seguintes objetivos específicos:

- (i) Investigar o estado da arte acerca de representação visual de linguagens de padrões de software;
- (ii) Desenvolver uma Linguagem de Padrões Ontológicos utilizando a notação visual usada até então em trabalhos do NEMO e identificar problemas dessa notação;
- (iii) Definir uma sintaxe abstrata que identifique os constructos semânticos para tratar aspectos estruturais e comportamentais de linguagens de padrões ontológicos;
- (iv) Definir uma sintaxe concreta para representação de aspectos estruturais e comportamentais de linguagens de padrões ontológicos;
- (v) Utilizar a notação proposta para representar uma Linguagem de Padrões Ontológicos desenvolvida no NEMO e analisar as melhorias obtidas em relação à notação utilizada anteriormente.

1.4 Método de Pesquisa

Este trabalho foi conduzido de acordo com os seguintes passos:

- i) *Revisão da Literatura*: neste passo ocorreu a aquisição de conhecimento sobre os temas relacionados ao trabalho, a saber: padrões, padrões ontológicos, linguagens de padrões, linguagens de padrões ontológicos e notações visuais. Inicialmente, foi realizada uma revisão informal da literatura, onde a pesquisa por publicações relacionadas foi realizada de forma não sistemática, tendo sido lidos artigos, livros, dissertações, teses e relatórios técnicos considerados relevantes ao trabalho. Nesse momento não houve restrições quanto ao uso de mecanismos de busca nem ao formato das publicações, bastando o material ter reconhecimento científico. Após a revisão informal, foi realizada uma investigação formal da literatura por meio de um mapeamento sistemático da literatura (KITCHENHAM; CHARTERS, 2007), quando foram investigadas e analisadas iniciativas envolvendo representação visual de Linguagens de Padrões de Software.

- ii) *Investigação do Estado da Prática*: neste passo ocorreu a análise da representação visual das LPOs existentes e a identificação de oportunidades de melhorias. LPOs foram objetos de análise em estudos experimentais envolvendo alunos do Programa de Pós-Graduação em Informática da UFES, que tiveram como objetivo avaliar a notação utilizada até então.
- iii) *Desenvolvimento da Abordagem Proposta*: neste passo foi elaborada a notação proposta neste trabalho. Para isso, foram considerados os achados do mapeamento sistemático, os resultados dos estudos experimentais realizados e os princípios de PoN (MOODY, 2009).
- iv) *Prova de Conceito*: neste passo foi realizada a reengenharia da representação visual da Linguagem de Padrões Ontológicos de Serviços (*Service Ontology Pattern Language - S-OPL*) (FALBO *et al.*, 2016), a fim de avaliar a viabilidade de uso da notação visual proposta e identificar as melhorias obtidas com a nova representação da LPO.
- v) *Avaliação da proposta*: neste passo foi realizado um *survey* com alunos do Programa de Pós-Graduação em Informática da UFES, a fim de avaliar a notação proposta considerando a perspectiva dos usuários.
- vi) *Escrita da Dissertação*: os resultados obtidos durante a execução dos passos anteriores foram documentados nesta dissertação.

Alguns dos resultados deste trabalho (ou relacionados a ele) foram publicados em eventos e periódicos, a saber:

- QUIRINO, G. K.; NARDI, J. C.; BARCELLOS, M. P.; *et al.* **Towards an Service Ontology Pattern Language**. Proceedings of the 34th International Conference on Conceptual Modeling (ER 2015). 2015. Stockholm, Sweden.
- LIVIERI, B.; GUARINO, N.; ZAPPATORE, M. S.; *et al.* **Ontology-based Modeling of Cloud Services: Challenges and Perspectives**. 8th IFIP WG 8.1 working conference on the Practice of Enterprise Modelling (PoEM 2015). 2015.
- RUY, F. B.; FALBO, R. DE A.; BARCELLOS, M. P.; *et al.* **An ISO-based software process ontology pattern language and its application for harmonizing standards**. ACM SIGAPP Applied Computing Review, v. 15, n. 2, p. 27–40, 2015.
- FALBO, R. DE A.; QUIRINO, G. K.; NARDI, J.; *et al.* **An Ontology Pattern Language for Service Modeling**. The 31st ACM/SIGAPP Symposium on Applied Computing. 2016. Pisa, Italy.

- TEIXEIRA, M. DAS G. DA S.; FALBO, R. DE A.; QUIRINO, G. K.; *et al.* **PoN-S: A Systematic Approach for Applying the Physics of Notation (PoN)**. Exploring Modelling Methods for Systems Analysis and Design Conference. 2016.

1.5 Organização da Dissertação

Neste capítulo inicial foram apresentadas as principais ideias desta dissertação, descrevendo o contexto de aplicação, motivações, objetivos e método de pesquisa. Além desta introdução, este texto é composto pelos seguintes capítulos e apêndices:

- **Capítulo 2 (Linguagens de Padrões Ontológicos e Notações Visuais):** aborda aspectos teóricos relacionados a LPOs e notações visuais relevantes a este trabalho.
- **Capítulo 3 (Notações Visuais em Linguagens de Padrões de Software: Mapeamento Sistemático):** apresenta os principais resultados do mapeamento sistemático realizado.
- **Capítulo 4 (Experimento para Avaliação de uma Linguagem de Padrões Ontológicos):** apresenta os resultados de um estudo experimental que teve como objetivo avaliar a notação utilizada até então para a representação de LPOs desenvolvidas no NEMO.
- **Capítulo 5 (Em Direção a uma Notação Visual para LPOs: Primeiros Passos):** apresenta S-OPL (*Service Ontology Pattern Language*) (FALBO *et al.*, 2016), uma LPO desenvolvida utilizando-se uma notação visual preliminar à notação definida neste trabalho e um estudo experimental realizado para avaliá-la.
- **Capítulo 6 (OPL-ML: Uma Notação Visual para Representação de Linguagens de Padrões Ontológicos):** apresenta a notação visual para LPOs proposta neste trabalho, chamada OPL-ML (*Ontology Pattern Language Modeling Language*). Também é apresentada a prova de conceito realizada, a qual consistiu na reengenharia da representação visual de S-OPL, e os resultados de um *survey* realizado para avaliar a notação visual proposta sob o ponto de vista dos usuários.
- **Capítulo 6 (Conclusão):** apresenta as considerações finais do trabalho, as contribuições e propostas de trabalhos futuros para continuidade e aprimoramento do trabalho.

- **Apêndice A (Formulários Utilizados nos Estudos Experimentais):**
apresenta os formulários utilizados nos estudos experimentais realizados.

Capítulo 6

Linguagens de Padrões Ontológicos e Notações Visuais

Neste capítulo são apresentados os principais conceitos relacionados a Linguagens de Padrões Ontológicos e Notações Visuais, temas centrais deste trabalho. Para isso, o capítulo encontra-se assim organizado: a Seção 2.1 apresenta uma breve introdução a ontologias; a Seção 2.2 aborda Padrões Ontológicos; a Seção 2.3 trata Linguagens de Padrões Ontológicos; a Seção 2.4 aborda notações visuais; e a Seção 2.5 apresenta as considerações finais do capítulo.

2.1 Ontologias

Uma ontologia é uma especificação formal e explícita de uma conceituação compartilhada, ou seja, um modelo abstrato que representa um fenômeno no mundo real (GRUBER, 1993). Em uma visão da comunidade de Web Semântica (DING, 2001), “conceituação” se refere a um modelo abstrato de uma realidade que identifica seus conceitos relevantes. “Explícita” significa que os conceitos usados e as restrições do seu uso são definidos explicitamente. A especificação é “formal”, pois é passível de entendimento por máquinas. A conceituação é “compartilhada” por capturar o conhecimento consensual aceito por uma comunidade.

Guizzardi (2007) faz uma importante distinção entre ontologias como modelos conceituais, conhecidas como *ontologias de referência*, e ontologias como artefatos de codificação, chamadas de *ontologias operacionais*. Uma ontologia de referência é construída com o objetivo de fazer a melhor descrição possível do domínio na realidade. É um tipo especial de modelo conceitual, um artefato de engenharia com o requisito adicional de representar um modelo de consenso dentro de uma comunidade. Por outro lado, uma vez que os usuários já tenham acordado uma concepção comum, as versões operacionais de uma ontologia de referência podem ser criadas. Ao contrário das ontologias de referência, ontologias operacionais são projetadas com o foco na garantia de propriedades computacionais desejáveis (FALBO *et al.*, 2013).

Ontologias possuem diversas classificações. Uma das mais conhecidas foi proposta por Guarino (1998), que define quatro tipos de ontologias com base no grau de generalidade: Ontologias de Fundamentação, Ontologias de Domínio, Ontologias de Tarefa e Ontologias de Aplicação. Ontologias de Fundamentação descrevem conceitos muito gerais como espaço, tempo, objeto, evento, ação etc. Elas servem como base para as ontologias de domínio e de tarefa. Ontologias de Domínio, por sua vez, descrevem o vocabulário relacionado a um domínio específico, como medicina ou automóveis. Ontologias de Tarefa

são aquelas capazes de descrever o vocabulário relacionado a uma tarefa genérica, como diagnose ou venda. Por fim, Ontologias de Aplicação descrevem conceitos dependentes de um domínio e uma tarefa particulares, as quais são, frequentemente, especializações de outras ontologias.

Scherp *et al.* (2011) também classificam ontologias de acordo com sua generalidade, mas enfocam a perspectiva estrutural, acrescentando um nível entre ontologias de fundamentação e ontologias de domínio, denominado ontologia de núcleo (*core ontology*). Uma ontologia de núcleo fornece uma definição precisa do conhecimento estrutural em uma área específica que cobre diferentes domínios de aplicação. São construídas baseadas em ontologias de fundamentação e representam um refinamento dessas, adicionando conceitos e relações específicos da área considerada.

Para Falbo *et al.* (2013), a variação de generalidade entre as ontologias pode ser vista como uma linha contínua, variando de ontologias de fundamentação para ontologias de domínio. De acordo com os autores, pode haver diferentes níveis de generalidade nas ontologias dentro do modelo que estão classificadas. Assim, as categorias de ontologias (Fundamentação, Núcleo e Domínio) podem ser vistas como regiões em um espectro com limites difusos entre eles. A Figura 2.1 ilustra essa visão contínua.

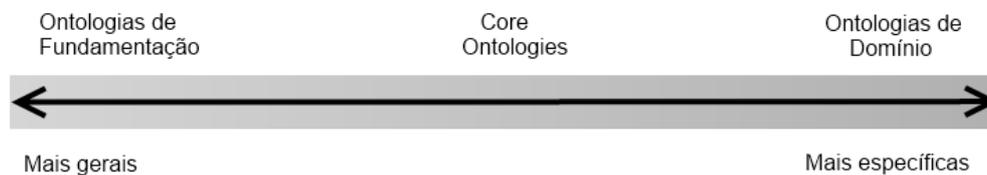


Figura 6.1: Níveis de generalidade de ontologias (adaptado de (FALBO *et al.*, 2013)).

2.2 Padrões Ontológicos

Atualmente, um engenheiro de ontologias que deseja desenvolver uma ontologia reutilizando outra, tipicamente, realiza buscas em um conjunto de ontologias leves e as avalia (usualmente de forma implícita) em relação à tarefa ou domínio de interesse ou utiliza ontologias de referência ou de núcleo que deveriam poder ser diretamente reutilizadas e especializadas. No primeiro caso, se alguma ontologia é selecionada, 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.

Devido às limitações no apoio à reutilização, o custo do reuso pode ser mais alto do que desenvolver uma ontologia a partir do zero (GANGEMI; PRESUTTI, 2009).

Nesse contexto, o uso de Padrões Ontológicos (POs) pode favorecer o reuso de experiências e boas práticas. POs descrevem um problema de modelagem recorrente originado no contexto de desenvolvimento de ontologias e apresentam uma solução comprovada para esse problema (FALBO *et al.*, 2013).

Existem muitos tipos diferentes de POs que podem ser utilizados em diferentes fases do processo de engenharia de ontologias. Neste trabalho, o interesse é em Padrões Ontológicos Conceituais (POCs). POCs são fragmentos de ontologias de fundamentação (POs de fundamentação - POFs) ou ontologias núcleo / domínio (POs relacionados a Domínio - PODs). Eles devem ser utilizados durante a fase de modelagem conceitual da ontologia e devem se concentrar apenas em aspectos conceituais, sem qualquer preocupação com a tecnologia ou linguagem a ser usada para implementação da ontologia (FALBO *et al.*, 2016).

Ruy *et al.* (2015c) discutem como POFs e PODs podem ser derivados. Segundo eles, PODs devem capturar o conhecimento básico relacionado a um domínio e, assim, eles podem ser vistos como fragmentos de uma ontologia de núcleo desse domínio. A complexidade dos PODs pode variar muito, dependendo da parte do domínio que está sendo representada. Às vezes, um POD contém apenas dois conceitos relacionados; em outras situações podem conter uma combinação complexa de conceitos e relações. É importante ressaltar que a mesma parte do domínio pode dar origem a dois (ou mais) padrões variantes ou padrões alternativos. Além disso, por vezes, um POD é estruturalmente aberto, a fim de ser completado por outro POD.

Ao extrair PODs de ontologias de núcleo ou de domínio, primeiro analisam-se os aspectos de domínio. A principal regra para um POD é representar um fragmento recorrente no domínio, independentemente da sua estrutura de fundamentação. Assim, enquanto POFs tendem a ser geralmente aplicados isoladamente, PODs de um domínio específico são muito interligados. Por essa razão, é comum aplicar PODs em combinação para engenharia de ontologias de domínio (RUY *et al.*, 2015c).

Para favorecer a reutilização dos PODs, é necessário organizá-los de modo a prover um mecanismo que auxilie essas combinações. Em um catálogo convencional, falta um senso forte de conexão. Mesmo um catálogo bem categorizado não provê ajuda suficiente para praticantes. Uma boa categorização ajuda o usuário a identificar um conjunto de padrões que são aplicáveis para especificar um problema. Entretanto, a categorização tende a falhar no

que se refere a guiar o usuário na seleção do padrão correto (ou sequência de padrões corretos) de um conjunto de padrões relacionados na mesma categoria (HAFIZ *et al.*, 2012). Assim, precisa-se de algo mais forte do que simplesmente saber que um outro padrão na coleção está relacionado de alguma forma a outro. Quando coleções são apresentadas em conjunto indicando, por exemplo, sequências de padrões, tem-se um forte senso de conexão. Isso é especialmente importante para a reutilização de PODs (FALBO *et al.*, 2014). Nesse sentido, Linguagens de Padrões Ontológicos (LPO) podem ser usadas para organizar os padrões e, diferente dos catálogos, elas apresentam os padrões em um processo de aplicação sistemático e guiado que explora as relações entre os padrões e favorece sua reutilização.

2.3 Linguagens de Padrões Ontológicos

Uma Linguagem de Padrões Ontológicos (LPO) é uma rede de padrões ontológicos inter-relacionados que fornece suporte para resolver problemas de desenvolvimento de ontologias para um domínio específico (FALBO *et al.*, 2013). Uma LPO fornece um forte senso de conexão entre os PODs, expressando vários tipos de relações entre eles, tais como relações de dependência, precedência temporal da aplicação e exclusão mútua entre padrões.

Uma LPO fornece orientação explícita sobre como reutilizar e integrar os padrões relacionados para se obter o modelo conceitual de uma ontologia concreta. Portanto, uma LPO é mais do que um catálogo de padrões. Ele inclui, além dos próprios padrões, um processo que guia o engenheiro de ontologias na seleção dos padrões a fim de aplicá-los de acordo com os problemas a serem modelados (FALBO *et al.*, 2016).

Para garantir uma aplicação estável e harmoniosa dos padrões, eles são apresentados em uma ordem de aplicação sugerida. Uma LPO é estruturada para apoiar e encorajar a aplicação de um padrão de cada vez, na ordem definida pelo processo da LPO e em função dos caminhos escolhidos pelo engenheiro de ontologias (FALBO *et al.*, 2013). Dessa forma, de acordo com os problemas a serem modelados, o engenheiro de ontologias é guiado a aplicar certos padrões, o que contribui para a produtividade no desenvolvimento da ontologia e para a qualidade do modelo da ontologia resultante (RUY *et al.*, 2015b).

Em resumo, um LPO dá orientações concretas e bem fundamentadas para o desenvolvimento de ontologias em um determinado domínio, abordando, no mínimo, as seguintes questões (FALBO *et al.*, 2013): (i) Quais são os principais problemas para resolver no domínio de interesse? (ii) Em que ordem esses problemas devem ser tratados? (iii) Que alternativas existem para resolver um determinado problema? (iv) Como as dependências

entre os problemas devem ser tratadas? (v) Como resolver cada problema individual de forma mais eficaz na presença dos problemas que o cercam?

No Núcleo de Estudos em Modelagem Conceitual e Ontologias (NEMO), têm sido desenvolvidas LPOs para aplicações em domínios diversos. Até o momento, foram desenvolvidas as seguintes LPOs: SP-OPL – *Software Process Ontology Pattern Language* (FALBO *et al.*, 2013), M-OPL - *Measurement Ontology Pattern Language* (BARCELLOS *et al.*, 2014), E-OPL – *Enterprise Ontology Pattern Language* (FALBO *et al.*, 2014), ISP-OPL – *ISO-based Software Process Ontology Pattern Language* (RUY *et al.*, 2015a) e S-OPL – *Service Ontology Pattern Language* (FALBO *et al.*, 2016).

Essas LPOs foram representadas por meio de diagramas de atividades da UML adaptados. O diagrama de atividades da UML é a notação UML padrão para representar restrições de sequenciamento temporais entre tipos de atividade e, conseqüentemente, para especificar a possível ordem de execução entre as atividades (FALBO *et al.*, 2013). Embora tenham sido utilizadas adaptações do diagrama de atividades da UML para representar as LPOs desenvolvidas, diferentes adaptações foram feitas para tratar cada LPO, resultando em uma falta de uniformidade nas notações visuais adotadas.

2.3.1. Um Exemplo de Linguagem de Padrões Ontológicos

Como um exemplo de LPO, nesta seção é apresentado um fragmento de E-OPL (FALBO *et al.*, 2014) e de sua aplicação na construção de uma ontologia organizacional para universidades brasileiras. Os aspectos tratados por E-OPL são: Arranjo Organizacional, Definição de Equipes, Papéis Institucionais, Objetivos Institucionais e Gerência de Recursos Humanos. A Figura 2.2 apresenta o diagrama que define o processo de E-OPL. Em seguida, um fragmento da descrição do processo é apresentado.

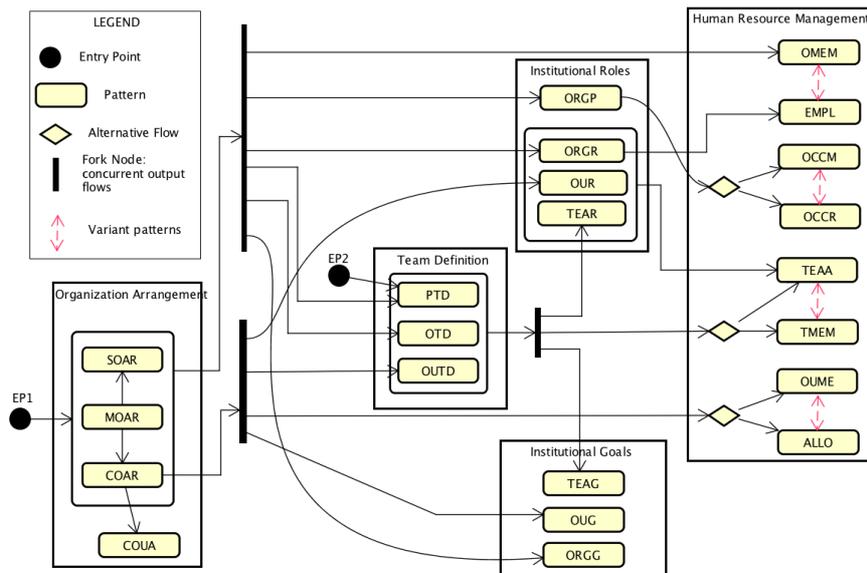


Figura 6.2: Diagrama de Processo de E-OPL.

E-OPL tem dois pontos de entrada. O engenheiro de ontologias deve escolher um deles, dependendo do escopo da ontologia organizacional específica a ser desenvolvida. Quando os requisitos para a nova ontologia organizacional incluem apenas problemas relacionados à definição de equipes de projetos, o ponto de entrada é EP2. Caso contrário, o ponto de partida é EP1. Nesse caso (EP1), primeiro o engenheiro de ontologias deve abordar os problemas relacionados à forma como uma organização é estruturada. Um dos três padrões seguintes devem ser selecionado: SOAR, COAR ou MOAR.

SOAR (*Simple Organization Arrangement*) deve ser selecionado como primeiro padrão se o engenheiro de ontologias necessita apenas representar organizações simples, que não são compostas por outras organizações ou unidades organizacionais. COAR (*Complex Organization Arrangement*) deve ser selecionado como o primeiro padrão se o engenheiro de ontologias necessita representar apenas organizações complexas, que não são compostas por outras organizações, mas são compostas por unidades organizacionais. Quando COAR é selecionado, COUA (*Complex Organizational Unit Arrangement*) pode ser usado na sequência, se existe a necessidade de representar unidades organizacionais complexas, que são compostas por outras unidades organizacionais. Finalmente, MOAR (*Multi-Organization Arrangement*) deve ser selecionado como primeiro padrão se o engenheiro de ontologias necessita representar organizações que são compostas por outras organizações. Quando MOAR é usado, o engenheiro de ontologias pode também usar SOAR e COAR para tratar a estrutura organizacional das organizações que compõem uma multiorganização.

Uma vez que os problemas relacionados ao arranjo organizacional foram resolvidos, o engenheiro de ontologias pode tratar problemas relacionados à definição de equipes, objetivos e papéis organizacionais e alguns problemas relacionados ao gerenciamento de recursos humanos. O processo referente ao uso dos padrões que tratam esses problemas não é descrito nesta seção e pode ser encontrado em (FALBO *et al.*, 2014).

Cada padrão presente na LPO é registrado em uma tabela. A Tabela 2.1 apresenta, como exemplo, o registro de alguns padrões de E-OPL.

Tabela 6.1 – Exemplo de registro de padrões presentes em uma LPO.

Id	Nome	Descrição
SOAR	Arranjo Organizacional Simples (<i>Simple Organization Arrangement</i>)	Define organizações simples, ou seja, que não são compostas por unidades organizacionais ou por outras organizações.
COAR	Arranjo de organizações complexas (Complex Organization Arrangement)	Define organizações complexas, que não são compostas por outras organizações, mas são compostas por unidades organizacionais.
COUA	Arranjo de unidades organizacionais complexas (Complex Organizational Unit Arrangement)	Define unidades organizacionais complexas, que são compostas por outras unidades organizacionais
MOAR	Arranjo de multiorganizações (Multi-Organization Arrangement)	Define organizações que são compostas por outras organizações

Para cada padrão é definida uma documentação, incluindo: nome, intenção, *rationale*, questões de competência, modelo conceitual e axiomatização.

A Figure 2.3 mostra os modelos conceituais de quatro padrões relacionados aos arranjos organizacionais. Os estereótipos mostrados nesse modelo são definidos em OntoUML. OntoUML é um perfil UML que permite que os modeladores façam distinções de modelagem minuciosas entre diferentes tipos de classes e relações de acordo com distinções ontológicas apresentadas pela Ontologia de Fundamentação Unificada (*Unified Foundational Ontology* - UFO), parte A (UFO-A) (GUIZZARDI, 2005). Note que, em E-OPL, os padrões são apresentados separadamente. Entretanto, quando combinados, eles formam modelos consistentes.

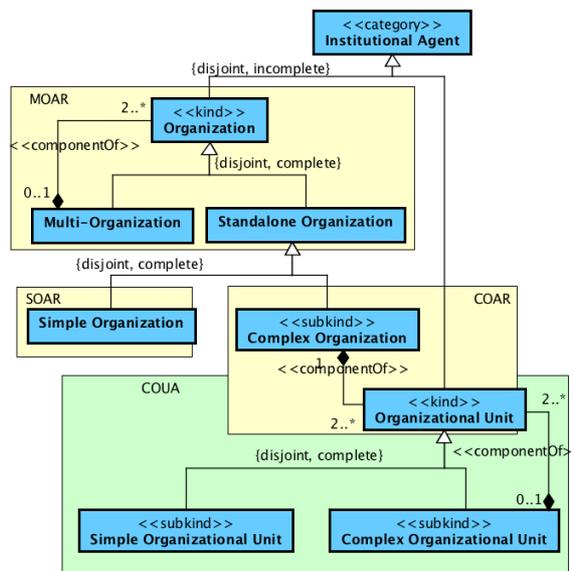


Figura 6.3 - Padrões sobre Arranjo Organizacional.

Para aplicar E-OPL, o engenheiro de ontologias deve seguir o processo e aplicar os padrões de acordo com os problemas a serem tratados. E-OPL foi utilizada em um estudo de caso com o intuito de desenvolver uma ontologia organizacional para universidades brasileiras, a qual trata aspectos relacionados à estrutura da universidade em relação às unidades envolvidas com a educação, aos papéis e posições envolvidos nesse contexto e aos membros universitários que exercem essas funções e ocupam esses cargos. A descrição completa da aplicação de E-OPL no desenvolvimento da ontologia organizacional para universidades brasileiras é apresentada em (FALBO et al., 2013). Aqui, será discutida apenas a aplicação dos padrões relacionados a arranjo organizacional.

Uma vez que há interesse em descrever a estrutura universitária, deve-se iniciar o processo de aplicação de padrões de E-OPL através do ponto de entrada EP1. As universidades são organizações autônomas, independentes hierarquicamente e organizadas em centros e departamentos. Assim, o padrão **COAR** deve ser selecionado como o primeiro padrão para ser aplicado. Dessa forma, Universidade (*University*) é considerada um subtipo de Organização Complexa (*Complex Organization*). Em seguida, para modelar unidades organizacionais e detalhar o arranjo organizacional de universidades brasileiras, deve-se aplicar o padrão **COUA**. Aplicando-se o padrão, torna-se possível modelar Centro Universitário (*University Center*), que é um subtipo de Unidade Organizacional Complexa (*Complex Organizational Unit*) e é composto por Departamentos (*Departaments*), um subtipo de Unidade Organizacional Simples (*Simple Organizational Unit*). A Figura 2.4 ilustra o fragmento da ontologia organizacional para universidades brasileiras referente à aplicação dos padrões **COAR** e **COUA**. Na figura é possível perceber os conceitos oriundos

dos padrões aplicados (em azul) e os conceitos deles especializados (em branco) para modelar aspectos específicos de organizações que são universidades brasileiras.

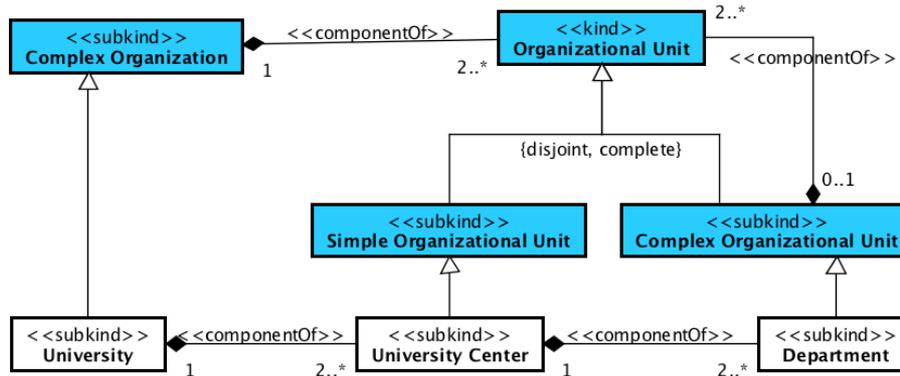


Figura 6.4 – Fragmento da Ontologia de Universidades produzida aplicando-se E-OPL (FALBO et al., 2013).

2.4 Notações Visuais

Notações visuais codificam as informações usando arranjos espaciais de elementos gráficos. São usadas em diversas áreas para a criação de representações gráficas que servem como abstrações capazes de auxiliar no entendimento das informações que se deseja comunicar. Segundo Moody (2009), notações visuais são efetivas porque elas são processadas em paralelo pelo sistema visual humano. Cerca de um quarto do cérebro humano é dedicado à visão, que é mais que todos os outros sentidos combinados. Além disso, diagramas e outras representações gráficas podem transmitir informações de forma mais concisa e precisa do que a linguagem comum.

No âmbito da Engenharia de Software notações visuais são fundamentais e desempenham papel particularmente importante na comunicação com os usuários finais e clientes, uma vez que se acredita que elas transmitam informações de forma mais eficaz para pessoas não-técnicas do que o texto. Apesar disso, historicamente, questões relacionadas à notação visual não têm tido seu valor reconhecido como deveriam, sendo comum dedicar os principais esforços em aspectos semânticos, deixando as convenções gráficas para uma reflexão tardia (MOODY, 2009).

Notações visuais também podem ser chamadas de representações visuais. Pode-se considerar que uma representação visual é formada por uma coleção de objetos e as relações entre esses objetos. O tipo de uma representação em particular é determinado pelas características dos símbolos usados para expressar esses objetos e relações (GURR, 1999).

Antes de se desenvolver uma notação visual é preciso determinar as características que se espera para a notação, tais como simplicidade e expressividade, que são comumente

desejadas em uma notação visual (MOODY, 2009). No entanto, características como essas são consideradas vagas e subjetivas. Uma característica mais objetiva e científica é a eficácia cognitiva, que se refere à velocidade, facilidade e precisão com que uma representação pode ser entendida (BOONE *et al.*, 2014).

Moody (2009), em seu trabalho conhecido como a Física das Notações (*Physics of Notations* - PoN), estabelece nove princípios para a criação de notações visuais cognitivamente eficazes, a saber:

- P1. Clareza Semiótica: Cada constructo semântico deve ser representado por exatamente um símbolo gráfico e vice-versa. Quatro tipos de anomalias podem ocorrer em uma notação: redundância de símbolos (um constructo semântico é representado por vários símbolos), sobrecarga de símbolos (um símbolo representa mais do que um constructo semântico), excesso de símbolos (o símbolo é criado e não representa qualquer constructo semântico) e déficit de símbolos (não há nenhum símbolo previsto para um certo constructo semântico).
- P2. Discriminabilidade Perceptiva: Símbolos diferentes devem ser facilmente diferenciados uns dos outros. A forma do símbolo é fator fundamental na distinção entre símbolos e, por conseguinte, deve ser usada como a variável visual primária para distinção entre diferentes constructos semânticos. Além da diferenciação pela forma, é indicado usar mais variáveis visuais, como cor, brilho e textura. Textos também podem ser utilizados.
- P3. Transparência Semântica: A representação de um constructo deve sugerir o seu significado. Uma maneira de projetar símbolos com transparência semântica é usando ícones, que levam a um reconhecimento mais rápido e recordação dos constructos. Além disso, eles melhoram a compreensão da notação principalmente para os usuários iniciantes. Representações semanticamente transparentes reduzem a carga cognitiva, porque elas têm mnemônicos embutidos e o seu significado pode ser percebido diretamente ou facilmente aprendido. Um símbolo é *semanticamente imediato* se um leitor iniciante é capaz de inferir o seu significado por si só a partir da sua aparência (por exemplo, um boneco para representar uma pessoa). Um símbolo é *semanticamente opaco* (ou *convencional*) se existe uma relação puramente arbitrária entre a sua aparência e seu significado (por exemplo, retângulos para representar entidades nos diagramas de entidades e relacionamentos). Um símbolo é *semanticamente perverso* (ou um *falso*

mnemônico), se um leitor iniciante é susceptível a inferir um significado diferente ou até mesmo oposto ao significado do símbolo.

P4. Gestão de Complexidade: É necessário ter mecanismos para lidar com a complexidade dos diagramas. Complexidade diagramática é medida pelo número de elementos de um diagrama. A complexidade pode ser reduzida de duas maneiras: o diagrama pode ser dividido em subdiagramas menores (modularização) ou os diagramas podem ser hierarquicamente estruturados para limitar os níveis de detalhes.

P5. Integração Cognitiva: Esta propriedade apenas se aplica quando múltiplos diagramas são usados. A notação deve ter mecanismos que permitam integrar informações de diferentes diagramas, sejam eles do mesmo tipo (integração homogênea) ou de tipos diferentes (integração heterogênea).

P6. Expressividade Visual: Deve-se utilizar toda a gama necessária de variáveis visuais para a expressividade da notação. A expressividade visual é determinada pelo número de variáveis visuais usadas em uma notação e a extensão em que elas são utilizadas, ou seja, refere-se à diversidade do vocabulário visual como um todo. A cor é um forte mecanismo para aumentar a expressividade visual de uma notação, uma vez que o contraste de cor é visto mais rápido do que as diferenças de outras variáveis. No entanto, ela só deve ser utilizada de forma redundante, porque as diferenças desaparecem quando diagramas são impressos em escala de cinza.

P7. Codificação Dupla: Textos devem ser utilizados como suplemento aos gráficos. Usar textos e gráficos juntos contribui para uma transmissão de informação mais efetiva do que usá-los separados. Entretanto, é importante que os símbolos sejam distinguíveis com base nas ilustrações. Rótulos (*labels*) podem ser usados para distinguir entre instâncias de símbolos, mas não entre tipos.

P8. Economia Gráfica: O número de tipos de símbolos em uma notação deve ser limitado e gerenciável. Este princípio pode ser adotado de três maneiras: constructos semânticos podem ser removidos; déficit de símbolos pode ser introduzido, mas isso prejudica a clareza semiótica da notação; e expressividade visual pode ser incrementada. Manipulação de múltiplas variáveis visuais reduz a necessidade de diminuir a quantidade de símbolos.

P9. Ajuste cognitivo: Diferentes dialetos visuais podem ser requeridos para diferentes tarefas ou públicos. Notações cognitivamente eficazes para os iniciantes podem

não ser cognitivamente eficazes para especialistas e vice-versa. Este princípio afirma, portanto, que variações nas notações podem ser necessárias para diferentes públicos ou tarefas.

Vale destacar que esses princípios interagem entre si. Essas interações possibilitam a ocorrência de conflitos (onde os princípios divergem uns com os outros) e a exploração de sinergias (onde princípios apoiam uns aos outros). A Figura 2.5 apresenta essas interações entre os princípios de PoN.

	Semiotic Clarity	Perceptual Discriminability	Semantic Transparency	Complexity Management	Cognitive Integration	Visual Expressiveness	Dual Coding	Graphic Economy	Cognitive Fit
Semiotic Clarity							±		
Perceptual Discriminability					+			+	
Semantic Transparency	+						±		
Complexity Management							-	+	
Cognitive Integration	-			+			-		
Visual Expressiveness		+					+	±	
Dual Coding								+	
Graphic Economy		+	+	-				+	
Cognitive Fit									

Figura 6.5 – Interações entre princípios: + indica um efeito positivo, - indica um efeito negativo, ± indica um efeito positivo ou negativo dependendo da situação (MOODY, 2009).

Os princípios estabelecidos por Moody (2009) definem propriedades desejáveis e mensuráveis para notações visuais e, assim, fornecem uma base para projeto e avaliação de notações. Melhorar uma notação visual com respeito a qualquer um dos princípios contribui para o aumento da sua eficácia cognitiva. Exemplos de avaliação de notações visuais utilizando-se os princípios podem ser encontrados em (MOODY *et al.*, 2010) e (BOONE *et al.*, 2014).

Neste trabalho utilizou-se dos princípios da PoN para projetar a notação visual de LPOs proposta. O desenvolvimento dessa notação foi guiado por *PoN-Systematized* (PoN-S) (TEIXEIRA *et al.*, 2016), uma abordagem sistemática que lida com a teoria e a prática da PoN em projetos de sintaxes concretas para linguagens visuais de modelagem. Essa abordagem é apresentada a seguir.

2.4.1. *PoN-Systematized (PoN-S)*

PoN-S (TEIXEIRA *et al.*, 2016) apresenta um processo formado por um conjunto ordenado de tarefas que guia o usuário na aplicação dos princípios de PoN durante o *design*

Na segunda fase, **Projetar Dialeto** (*Design Dialect*), cada dialeto identificado deve ter o seu conjunto de símbolos definidos de acordo com os objetivos e as diretrizes previamente identificados. Em outras palavras, nessa fase é produzida a sintaxe concreta da notação visual. Esta fase consiste em: (i) **Definir o Conjunto de Símbolos do Dialeto** (*Define the Dialect Symbol Set*), que é responsável pela definição de um símbolo para cada um dos elementos da sintaxe abstrata; e (ii) **Identificar Formas de Gerenciar a Complexidade do Modelo** (*Identify Ways to Manage Model Complexity*), que é opcional e realizada quando a quantidade de elementos exige gestão de complexidade do modelo. Para realizar essas atividades, o dialeto e a sintaxe abstrata devem ser considerados. A Figura 2.7 apresenta a atividade **Definir o Conjunto de Símbolos do Dialeto** (*Define the Dialect Symbol Set*) em detalhes.

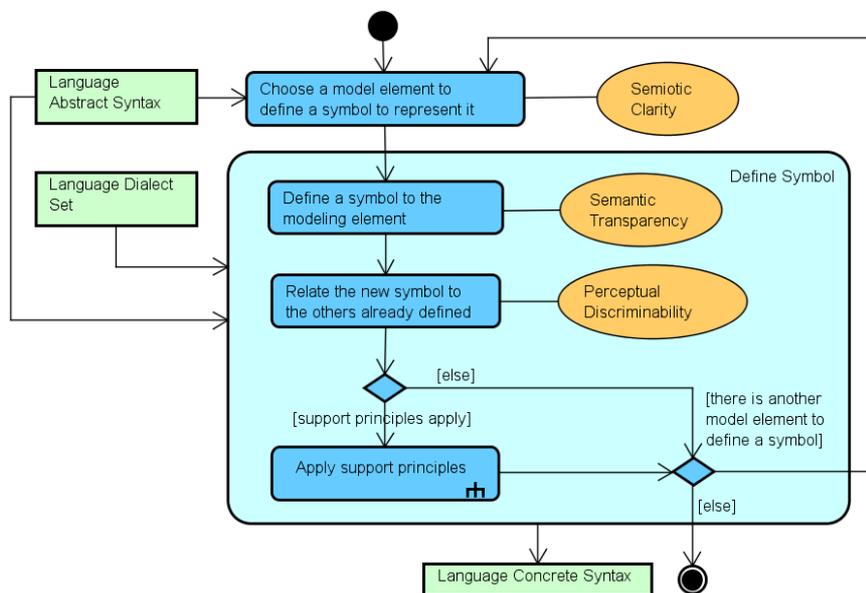


Figura 6.7 – Atividade Definir o Conjunto de Símbolos do Dialeto (TEIXEIRA et al., 2016).

A definição do conjunto de símbolos começa pela escolha de um elemento da sintaxe abstrata a ser representado que é realizada na atividade **Selecionar Elemento do Modelo para Representá-lo** (*Choose a Model Element to define a Symbol to Represent it*). O princípio da Claridade Semiótica deve ser considerado nesse momento, com o intuito de garantir que cada elemento será representado por exatamente um símbolo, a menos que uma exceção seja necessária devido às diretrizes estabelecidas para o dialeto. Uma vez que o elemento a ser representado é escolhido, deve-se **Definir um Símbolo para o Elemento de Modelagem** (*Define a Symbol to the Modeling Element*), considerando-se o princípio de Transparência Semântica, a fim de estabelecer um significado claro para o símbolo. Deve-se ainda, **Relacionar o Novo Símbolo a Outros já Definidos** (*Relate the New Symbol to the Others Already Defined*) na sintaxe concreta e analisar a distância visual entre eles, seguindo o princípio

da Discriminabilidade Perceptiva. Essas atividades são realizadas num ciclo até que todos os elementos da representação desse dialeto sejam definidos.

Após a definição dos símbolos de um dialeto, o projetista deve decidir se é necessário gerenciar a complexidade dos modelos gerados pelo dialeto. A atividade **Aplicar os Princípios de Suporte** (*Apply Support Principles*), que faz uma análise dos elementos quanto aos princípios Expressividade Visual, Economia Gráfica e Codificação Dupla, é uma atividade opcional, cuja importância aumenta à medida que a notação cresce em tamanho e complexidade.

A Figura 2.8 detalha a atividade **Identificar Formas de Gerenciar a Complexidade do Modelo** (*Identify Ways to Manage Model Complexity*), na qual, a partir das sintaxes abstratas e concretas e das características do dialeto, são definidas estratégias de representação para gerenciar a complexidade do modelo (quantas o projetista considerar necessário).

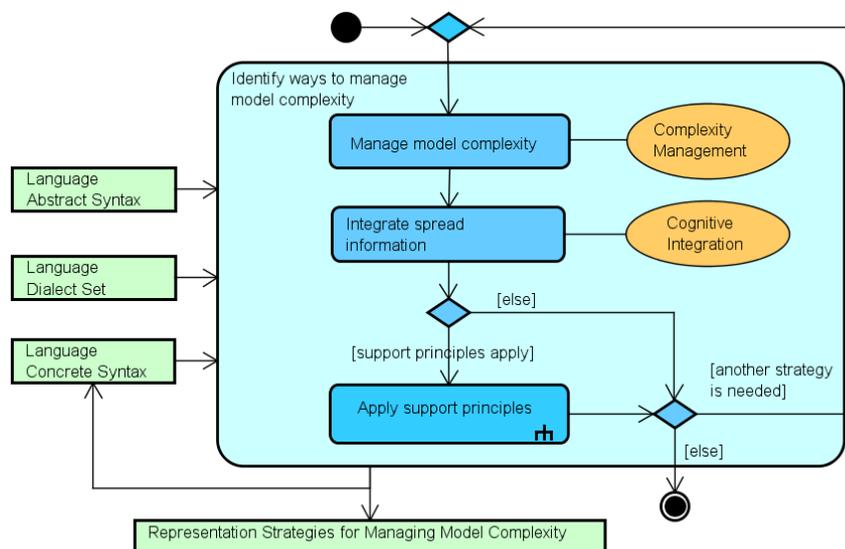


Figura 6.8 – Atividade Identificar Formas de Gerenciar a Complexidade do Modelo (TEIXEIRA et al., 2016).

A primeira atividade é **Gerenciar a Complexidade do Modelo** (*Manage Model Complexity*), que é guiada pelo princípio de Gestão de Complexidade para que sejam estabelecidas estratégias de representação para gerenciar a complexidade dos diagramas gerados pelo dialeto. Um exemplo é o uso de modularização. A segunda atividade, **Integrar Informações Espalhadas** (*Integrate Spread Information*), é guiada pelo princípio de Integração Cognitiva e é responsável por estabelecer formas de rastrear as informações espalhadas em diversos diagramas e definir estratégias para conectá-las. Essas duas atividades podem ser realizadas em paralelo, resultando em uma estratégia única de representação que esteja de acordo com ambos os aspectos da gestão de complexidade (organização e integração de

informações). Na verdade, ambas as atividades são aplicadas em um *loop* até ser considerado suficiente pelo projetista. Normalmente, cada ciclo resulta em uma estratégia de representação para gerenciar a complexidade do modelo que, se necessário, é complementado com novos elementos na sintaxe concreta.

A Figura 2.9 detalha a atividade **Aplicar os Princípios de Suporte** (*Apply Support Principles*), que é uma atividade opcional no *design* de um dialeto e lida com a aplicação dos três princípios de suporte: Expressividade Visual, Economia Gráfica e Codificação Dupla. O projetista pode aplicar cada princípio tanto quanto julgar necessário e não há uma ordem pré-definida a ser seguida. Considerando-se as sintaxes abstrata e concreta da notação visual e as características do dialeto, pode-se realizar alterações na sintaxe concreta ou em alguma estratégia da gerência de complexidade do modelo. A atividade **Melhorar a Utilização de Variáveis Visuais** (*Improve the use of Visual Variables*) é guiada pelo princípio Expressividade Visual. Nela, o projetista deve rever os símbolos (ou estratégias) e possivelmente atualizar os valores das variáveis visuais para maximizar a sua expressividade. O projetista pode fazer isso individualmente (por símbolo) ou considerando o conjunto de símbolos como um todo. A atividade **Simplificar o Conjunto de Símbolos** (*Simplify the Symbol Set*) é guiada pelo princípio da Economia Gráfica. Nela, o projetista também pode rever os símbolos (ou estratégias), agora com o objetivo de simplificar o dialeto. Finalmente, na atividade **Definir Complementos Textuais** (*Define Textual Complement*), aplicando-se o princípio de Codificação Dupla, o projetista deve avaliar quando é útil introduzir redundância por meio do uso de texto. Isso pode ser necessário quando o projetista considerar que o texto vai aumentar a expressividade do símbolo.

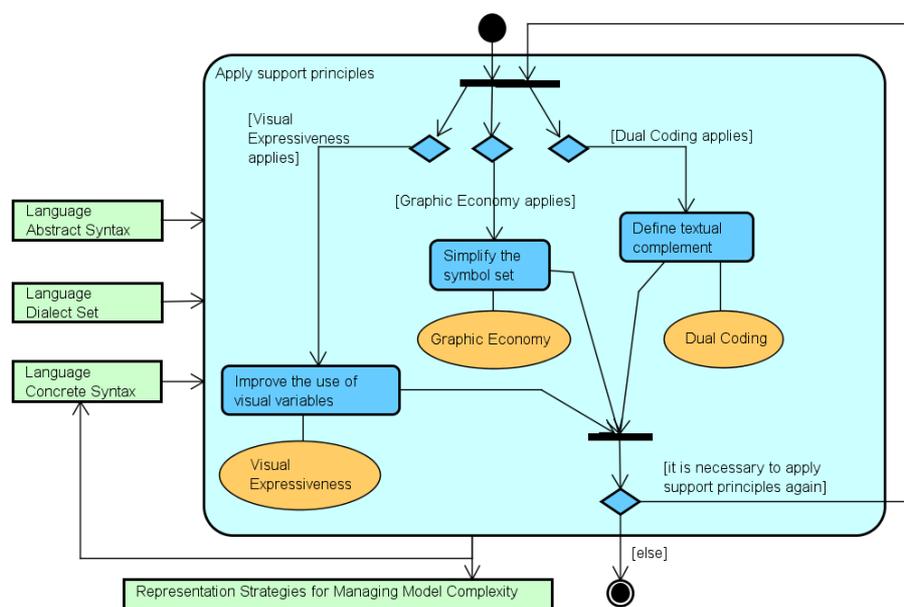


Figura 6.9 – Atividade Aplicar os Princípios de Suporte.

2.5 Considerações Finais

Para introduzir assuntos importantes ao entendimento deste trabalho, este capítulo abordou conteúdo relacionado a Linguagens de Padrões Ontológicos e Notações Visuais.

No que se refere a LPOs, foram abordados aspectos relacionados a ontologias, padrões ontológicos e LPOs propriamente ditas. Além disso, foi apresentado um exemplo de LPO. Em relação a Notações Visuais, os principais princípios da Física das Notações (*Physics of Notation* - PoN) foram apresentados e o processo de sistematização da PoN, *PoN-Systematized* (PoN-S), utilizado neste trabalho foi brevemente apresentado.

No próximo capítulo são apresentados os principais resultados de um mapeamento sistemático no qual foram investigadas notações visuais adotadas para representar Linguagens de Padrões de Software, a fim de se obter uma visão do estado da arte relacionado ao tópico de pesquisa explorado neste trabalho.

Capítulo 3

Notações Visuais em Linguagens de Padrões de Software: Mapeamento Sistemático

Neste capítulo são apresentados os principais resultados de um Mapeamento Sistemático (MS) que investigou notações visuais utilizadas para representar Linguagens de Padrões de Software. Uma vez que Linguagens de Padrões Ontológicos é um tema recente, decidiu-se por investigar linguagens de padrões em uma área onde seu uso é consolidado, a fim de se obter informações que sejam relevantes no contexto de notações visuais para Linguagens de Padrões Ontológicos. O capítulo encontra-se assim organizado: a Seção 3.1 apresenta uma breve introdução sobre Linguagens de Padrões e relacionamentos entre padrões; a Seção 3.2 apresenta o processo utilizado para conduzir o MS; a Seção 3.3 apresenta o protocolo utilizado no estudo; a Seção 3.4 descreve a execução do MS e faz uma síntese dos dados obtidos; a Seção 3.5 provê uma discussão sobre os dados obtidos e, na Seção 3.6, são apresentadas as considerações finais do capítulo.

3.1 Linguagens de Padrões

Segundo Deutsch *et al.* (2004), padrões são veículos para o encapsulamento de conhecimento e permitem capturar o que deve ser feito para resolver um dado problema. Para utilizar um padrão é necessário reconhecer a oportunidade de aplicá-lo, ou seja, é preciso entender e reconhecer o contexto em que o problema recorrente tratado pelo padrão ocorre e identificar se o problema que se deseja tratar é uma instância do problema recorrente. Para isso, padrões devem conter descrições para o problema, a solução e o contexto em que ocorrem.

Buschmann (2007) ressalta que muitos padrões encontrados na literatura são relacionados a outros, mas a maioria falha em explicar como os padrões podem ser combinados para formar soluções para problemas maiores do que os tratados por cada padrão. As descrições de soluções tendem a se concentrar em aplicar os padrões de forma isolada e não resolver adequadamente os problemas que surgem quando vários padrões são aplicados de forma combinada. Essa situação é problemática, uma vez que características introduzidas por um padrão podem ser requeridas por outros. Portanto, é necessário um contexto mais amplo para descrever os problemas maiores que podem ser resolvidos e os que podem surgir quando os padrões são usados de maneira combinada. Esse contexto é provido por uma Linguagem de Padrões.

No âmbito de Engenharia de Software, uma Linguagem de Padrões (LP) é uma rede de padrões inter-relacionados que define um processo para resolução sistemática de

problemas relacionados ao desenvolvimento de software (DEUTSCH, 2004). Esse processo tem como objetivo fornecer suporte global para a utilização dos padrões de modo a resolver problemas relacionados a um domínio técnico ou de alguma aplicação específica. Essa visão holística deve fornecer orientação explícita sobre os problemas que podem surgir no domínio, informar os possíveis meios de resolvê-los e sugerir um ou mais padrões para resolver cada problema específico (FALBO *et al.*, 2013).

As linguagens de padrões refletem o fato de que os padrões tendem a ser fortemente acoplados e é difícil, ou até mesmo impossível, utilizá-los de forma isolada (FALBO *et al.*, 2013). Nesse sentido, para que uma LP seja eficaz no seu objetivo de guiar o usuário na aplicação dos padrões e trate de maneira adequada a aplicação combinada de vários padrões, é necessário que as relações entre os padrões sejam definidas e explicitadas na LP.

Notações visuais podem ser utilizadas para representar graficamente LPs. O propósito da adoção de notações visuais é dar uma visão geral dos padrões e seus relacionamentos para prover o entendimento holístico da LP e auxiliar na seleção dos padrões.

Não há consenso sobre a notação visual a ser adotada para representar LPs. Como consequência, LPs podem adotar diferentes notações visuais. Buscando-se identificar e analisar notações visuais adotadas em LPs de software foi realizada uma investigação da literatura por meio de um mapeamento sistemático, o qual é apresentado nas próximas seções.

3.2 Método Adotado

Segundo (KITCHENHAM *et al.*, 2011), um mapeamento sistemático (também conhecido como estudo exploratório) realiza um amplo estudo em um tópico de tema específico e visa identificar evidências disponíveis sobre esse tema. Buscando-se assegurar um mapeamento imparcial, rigoroso e repetível, o estudo foi realizado seguindo-se o processo definido em (KITCHENHAM; CHARTERS, 2007) o qual envolve três atividades:

- i. *Desenvolver o Protocolo:* nesta atividade o pesquisador realiza a prospecção sobre o tema de interesse do estudo, definindo o contexto e o objeto de análise. Em seguida, o protocolo que será o guia para execução do estudo é definido, testado e avaliado. O protocolo deve conter todas as informações necessárias para executar a pesquisa (questões de pesquisa, critérios para seleção das fontes, critérios para seleção das publicações, procedimentos para armazenar e analisar os resultados, e assim por diante).

- ii. *Conduzir a Pesquisa*: nesta atividade o pesquisador executa o protocolo definido e, assim, seleciona, armazena e realiza análises quantitativas e qualitativas dos dados coletados.
- iii. *Relatar Resultados*: nesta atividade o pesquisador empacota os resultados gerados ao longo da execução do estudo e os publica em alguma conferência, revista, relatório técnico, biblioteca de trabalhos científicos ou outro veículo.

3.3 Protocolo de Pesquisa

Conforme discutido anteriormente, Linguagens de Padrões Ontológicos (LPOs) é um tema muito recente. Como consequência, além dos trabalhos desenvolvidos no NEMO, não foram encontrados trabalhos contendo representações visuais para LPOs. Dessa forma, considerando-se que Linguagens de Padrões é um tema consolidado na área Engenharia de Software, decidiu-se investigar as notações visuais adotadas para representar Linguagens de Padrões de Software.

O **objetivo** deste mapeamento sistemático é, portanto, identificar trabalhos existentes na literatura que propõem ou aplicam Linguagens de Padrões de Software que tenham alguma representação gráfica ou visual e analisar as representações adotadas.

Para obter informações para o alcance do objetivo estabelecido, foram definidas as **questões de pesquisa** (QPs) apresentadas na Tabela 3.1.

Tabela 3.1 – Questões de Pesquisa do Mapeamento Sistemático.

Id	Questão	Rationale
QP1	Quando e em quais tipos de veículos (periódico/evento científico) os trabalhos foram publicados?	Prover um panorama sobre quando e onde os trabalhos foram publicados, permitindo analisar a maturidade do tópico de pesquisa. Verificar se há períodos de maior ou menor quantidade de publicações e qual a distribuição dos trabalhos considerando eventos científicos e periódicos.
QP2	Quais tipos de pesquisas têm sido feitos?	Identificar os tipos de pesquisa considerando a classificação proposta por (WIERINGA <i>et al.</i> , 2006)
QP3	As LPs propostas são de propósito geral ou específico?	Investigar se as LPs foram definidas para uso independente de domínio ou classe de aplicações ou para domínios ou classes de aplicações específicos.
QP4	Quais tipos de padrões são encontrados nas linguagens de padrão propostas?	Identificar os tipos de padrões (por exemplo, padrões de arquitetura, <i>design patterns</i> etc.) presentes nas LPs.
QP5	Quais elementos são tratados na linguagem de padrões e que símbolos ou notações são utilizados para representá-los?	Identificar os elementos representados nas LPs (por exemplo, padrões e relações) e os símbolos adotados.

Tabela 3.1 – Questões de Pesquisa do Mapeamento Sistemático (cont.).

Id	Questão	Rationale
QP6	As LPs definem os tipos de relações nela existentes? Se sim, quais os tipos de relações definidos?	Identificar se as LPs definem explicitamente os tipos de relações representadas e quais são os tipos definidos.
QP7	Como padrões são agrupados nas LPs e como os agrupamentos são representados?	Investigar se há agrupamentos de padrões nas LPs, quais critérios para agrupamento são utilizados e como os grupos são representados.
QP8	Além da representação gráfica da linguagem, que mecanismo é fornecido para guiar a seleção dos padrões?	Investigar se as LPs proveem algum mecanismo (por exemplo, a descrição de um processo) guiando a seleção dos padrões.
QP9	Quais são os tipos de modelos providos pela LP (estrutural / processo / ambos)?	Identificar se as LPs fornecem modelos estruturais, que representam os elementos das LPs e as relações entre eles, modelos de processo, que representam fluxos que indicam os caminhos possíveis para seleção dos padrões, ou ambos.

A **expressão de busca** adotada no estudo é formada por dois grupos de termos conectados por um operador AND. O primeiro grupo visa capturar estudos relacionados a linguagens de padrões. O segundo grupo visa capturar estudos relacionados a software. Dentro do primeiro grupo, o operador OR foi utilizado permitindo a existência de termos alternativos. Após diversos testes com diversas expressões de busca distintas, a seguinte expressão de busca foi utilizada: (((*“Pattern Language”*) OR (*“Patterns Language”*)) AND (*“Software”*)).

As **fontes** utilizadas no estudo são sete bibliotecas digitais: Scopus (www.scopus.com), Engineering Village (www.engineeringvillage.com), ACM (dl.acm.org), IEEE Xplore (ieeexplore.ieee.org), Springer (link.springer.com), ScienceDirect (www.sciencedirect.com) e Web of Science (www.webofknowledge.com).

Os **objetos de análise** do estudo são artigos publicados em eventos científicos ou periódicos.

O **procedimento para seleção das publicações** compreende 4 etapas:

1ª etapa (E1) - Seleção e catalogação preliminar das publicações: A seleção preliminar das publicações consiste em aplicar a expressão de busca usando o mecanismo de busca das bibliotecas digitais selecionadas.

2ª etapa (E2) – *Remoção de duplicatas*: estudos indexados por mais de uma biblioteca digital são identificados e as duplicações removidas nesta etapa.

3ª etapa (E3) - *Seleção das publicações relevantes - 1º filtro*: A seleção das publicações através dos critérios de busca não garante que todas as publicações selecionadas sejam úteis no contexto da pesquisa, pois a aplicação dos critérios de busca é restrita ao aspecto sintático. Sendo assim, o resumo/*abstract* de cada publicação selecionada deve ser submetido à análise, sendo excluídas as publicações que não atenderem ao critério de inclusão CI1 ou atenderem a um dos critérios de exclusão CE1 e CE2, descritos a seguir:

CI1: A publicação propõe ou aplica uma linguagem de padrões de software.

CE1: O estudo não possui resumo (*abstract*).

CE2: A publicação não é um estudo primário.

4ª etapa (E4) - *Seleção das publicações relevantes - 2º filtro*: Uma vez que a seleção das publicações utilizando o 1º filtro considera a análise apenas do resumo/*abstract* da publicação, é possível que nem todas as publicações selecionadas contenham informações relevantes para o estudo. As publicações selecionadas com a aplicação do 1º filtro devem, então, ser submetidas a uma análise detalhada e, após leitura e análise do conteúdo completo de cada publicação, devem ser excluídas aquelas que não atenderem aos critérios de inclusão CI1 ou CI2 ou atenderem a algum dos critérios de exclusão CE3 a CE6 descritos a seguir:

CI2: A linguagem de padrões descrita na publicação possui algum tipo de representação visual, gráfica ou diagramática.

CE3: O artigo não está escrito em inglês.

CE4: A publicação é uma cópia ou uma versão antiga de uma publicação já selecionada.

CE5: O estudo foi publicado somente como um resumo (*abstract*).

CE6: O texto completo da publicação não está disponível.

3.4 Execução do Mapeamento e Síntese dos Dados

O mapeamento sistemático considerou artigos publicados até dezembro de 2015. Como resultado da etapa E1, 1159 publicações foram obtidas. Após as duplicações serem removidas (E2), restaram 583 publicações. Dessas, na etapa E3, foram selecionadas 250 publicações, as quais foram reduzidas a 54 em E4. A Figura 3.1 ilustra o processo realizado para a seleção das publicações, que resultou em 54 publicações selecionadas.

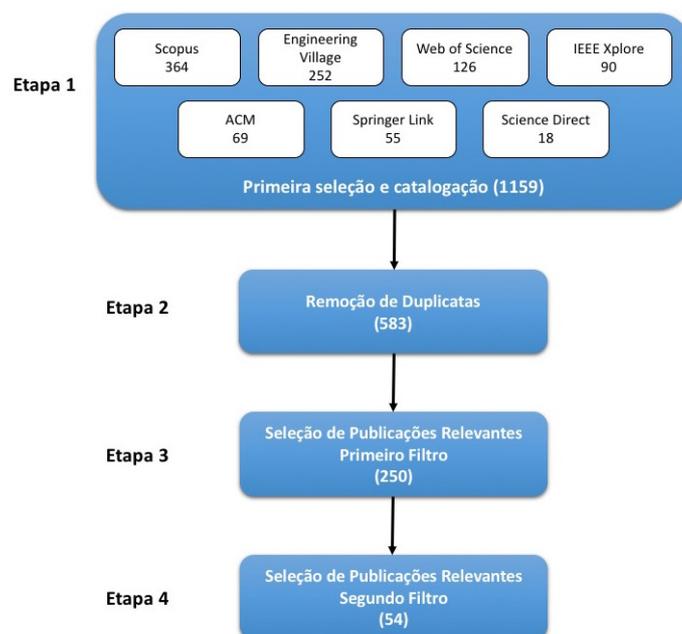


Figura 3.1 - Processo de Seleção das Publicações

Informações sobre as publicações selecionadas encontram-se na Tabela 3.2. Após a tabela, é apresentada uma síntese dos dados extraídos das publicações para cada questão de pesquisa.

Tabela 3.2 – Publicações selecionadas.

ID	Publicação	Descrição
01	(NOBLE, 1998)	Apresenta alguns padrões de projeto orientados a objetos e os organiza em uma linguagem de padrões.
02	(MAHEMOFF <i>et al.</i> , 2001)	Apresenta uma LP que inclui padrões de segurança e usabilidade e defende a aplicação desses padrões para sistemas onde a segurança é um ponto crítico. A LP é aplicada em um estudo de caso com um sistema de alarme.
03	(BORCHERS, 2001)	Apresenta padrões para interação humano-computador (<i>human-computer interaction</i> (HCI)) e dá dois exemplos de linguagens de padrões nesse contexto.
04	(GOEDICKE; ZDUN, 2002)	Apresenta uma linguagem de padrões para migração fragmentada de sistemas legados e a aplica em um estudo de caso.
05	(GZARA <i>et al.</i> , 2003)	Apresenta uma linguagem de padrões para <i>Product Information Systems</i> (PIS).
06	(GOEDICKE <i>et al.</i> , 2004)	Apresenta uma linguagem de padrões para a construção de linguagens de gerenciamento de pontos de variação, em tempo de execução, específicos de domínio. Além de apresentar a LP e os padrões, três casos de estudos são discutidos.
07	(HANMER; KOCAN, 2004)	O artigo apresenta a estrutura de linguagens de padrão e dá dois exemplos de LPs, apresentando seus diagramas e padrões.
08	(ZDUN, 2004a)	Apresenta uma linguagem de padrões para auxiliar em projetos que envolvem aspectos de composição de frameworks. Possíveis seqüências de uso dos padrões são exploradas e um <i>Feature Model</i> , foi desenvolvido para auxiliar na seleção dos padrões.

Tabela 3.2 – Publicações selecionadas (cont.).

ID	Publicação	Descrição
09	(ZDUN, 2004b)	Apresenta uma linguagem de padrões para implementação de abstrações de métodos dinâmicos e para a combinação destes com linguagens que não têm suporte a eles. Além da apresentação da PL e dos padrões, é apresentado um estudo de caso mostrando vários exemplos de utilização da linguagem de padrões.
10	(ZDUN <i>et al.</i> , 2004)	Propõe uma abordagem de Engenharia Orientada a Modelos (Model-Driven Engineering) para produzir um repositório de artefatos de modelagem que visa apoiar o desenvolvimento de Resource Constrained Embedded Systems (RCES). Entre os artefatos estão padrões de Security & Dependability, que são armazenados em um repositório, permitindo que as relações entre eles sejam capturadas por ferramentas e representadas visualmente em uma linguagem de padrões.
11	(BRAGA; MASIERO, 2004)	Propõe um processo para descoberta de <i>hot spots</i> em linguagens de padrões. O estudo de caso apresentado no artigo usa uma linguagem de padrões para Gestão de Recursos de Negócios
12	(MEISTER <i>et al.</i> , 2004)	Apresenta uma arquitetura de linha de produtos baseada em padrões para o domínio de softwares de análise estatística. Uma linguagem de padrões para projetos de linha de produtos de software analíticos é apresentada. A aplicação da associação entre arquitetura e a LP é explorada em dois exemplos.
13	(AVGERIOU; ZDUN, 2005)	Apresenta uma linguagem de padrões para organizar padrões arquiteturais.
14	(MOURATIDIS <i>et al.</i> , 2005)	Apresenta uma LP para segurança de software e sua formalização.
15	(WELLHAUSEN, 2005)	Apresenta uma linguagem de padrões para áreas de buscas de dados em interfaces de usuários.
16	(LUKOSCH; SCHÜMMER, 2006)	Apresenta uma linguagem de padrões que cobre questões de alto nível, bem como de baixo nível, de projetos colaborativos (<i>groupware</i>), permitindo organizar e dar suporte ao desenvolvimento coletivo de sistemas.
17	(AGUIAR; DAVID, 2006)	Apresenta uma linguagem de padrões para documentação de <i>frameworks</i> orientados a objetos.
18	(AVGERIOU; TANDLER, 2006)	Apresenta uma linguagem de padrões para sistemas colaborativos.
19	(GROLIMUND; MÜLLER, 2006)	Apresenta uma linguagem de padrões que contém padrões de arquitetura que tratam <i>overlay networks</i> para sistemas <i>peer-to-peer</i> .
20	(KHALED; HOSNY, 2006)	Apresenta uma linguagem de padrões que trata de aspectos relacionados ao desempenho de sistemas considerando três indicadores chave de desempenho: responsividade, estabilidade e manutenibilidade.
21	(BELLEBIA; DOUIN, 2006)	Apresenta uma linguagem de padrões para construir mediadores (<i>middlewares</i>) para sistemas embarcados. Além disso, essa LP é aplicada em um sistema WBJDP (<i>Web-Based JavaCard Development Platform</i>).
22	(MENG <i>et al.</i> , 2007)	Apresenta uma linguagem de padrões para Métodos Ágeis. A partir do uso combinado dos padrões pode-se definir processos de desenvolvimento de software que incluem práticas ágeis.
23	(KIRK <i>et al.</i> , 2007)	Apresenta estudos sobre reutilização de <i>frameworks</i> orientados a objetos. Dentre os aspectos investigados está o uso de linguagens de padrões como forma de contribuir para a reutilização desses <i>frameworks</i> . A linguagem de padrões apresenta Design Patterns que auxiliam no desenvolvimento e avaliação de Frameworks.

Tabela 3.2 – Publicações selecionadas (cont.).

ID	Publicação	Descrição
24	(MONTEIRO; AGUIAR, 2007)	Apresenta uma linguagem de padrões que tem como objetivo apoiar o refatoramento de sistemas existentes em novas versões orientadas a aspectos.
25	(AHLUWALIA, 2007)	Apresenta uma linguagem de padrões que pode ser usada no desenvolvimento de sistemas altamente escaláveis.
26	(HENTRICH; ZDUN, 2007)	Apresenta uma linguagem de padrões para arquiteturas orientadas a serviços e processos a fim de integrar serviços e processos de negócio através da invocação a serviços a partir das atividades dos processos de negócio.
27	(DELESSY <i>et al.</i> , 2007)	Apresenta uma linguagem de padrões voltada para protocolos de segurança.
28	(SANTOS; KOSKIMIES, 2008)	Apresenta uma linguagem de padrões para desenvolvimento de <i>frameworks</i> de reutilização de interfaces de alto nível.
29	(WEYNS, 2009)	Apresenta uma linguagem de padrões para projeto de arquitetura de sistemas multiagentes.
30	(JOHN <i>et al.</i> , 2009)	Apresenta uma linguagem de padrões composta por padrões arquitetônicos de apoio à usabilidade (USAPs - <i>Usability-Supporting Architectural Patterns</i>). Uma ferramenta de apoio ao uso da LP proposta também é apresentada.
31	(SALAH; ZEID, 2009)	Apresenta PLITS (<i>Pattern Language for Intelligent Tutoring Systems</i>), que é composta por padrões extraídos através de engenharia reversa de sistemas de Tutoria Inteligente já existentes.
32	(ELORANTA <i>et al.</i> , 2009)	Apresenta uma linguagem de padrões de arquitetura para sistemas de controle de máquinas.
33	(HENTRICH, 2009)	Apresenta uma linguagem de padrões para tratar problemas de sincronização ao projetar e implementar arquiteturas orientadas a serviços e processos.
34	(ZAMANI <i>et al.</i> , 2009)	Apresenta um processo para verificação da aplicação de uma linguagem de padrões de projeto composta por padrões de EAA (<i>Enterprise Application Architecture</i>) de Martin Fowler.
35	(SESERA, 2010)	Apresenta a aplicação de uma linguagem de padrões para desenvolvimento de sistemas bancários em alguns casos desse domínio.
36	(HANNEBAUER <i>et al.</i> , 2010)	Apresenta uma linguagem de padrões para desenvolvimento de softwares gratuitos, livre e <i>open source</i> .
37	(FOGLI <i>et al.</i> , 2010)	O artigo descreve alguns problemas e possíveis soluções para o desenvolvimento de serviços web governamentais. Para isso, utilizam duas PLs: “ <i>Pattern language for government service creation</i> ” e a “ <i>HCI pattern language for EUD of G2C services</i> ”, que adotam a mesma notação gráfica.
38	(HSIEH <i>et al.</i> , 2011)	Apresenta uma LP para desenvolvimento de software para múltiplas plataformas que fazem um extensivo uso de sistemas de integração contínua.
39	(AMATRIAIN; ARUMI, 2011)	Apresenta um processo de criação de <i>frameworks</i> de desenvolvimento de software. Uma das etapas do processo consiste em definir uma linguagem de padrões. O processo é aplicado no contexto de sistemas multimídia e como um dos resultados dessa aplicação foi definida uma Linguagem de Padrões para Sistemas de Processamento de Multimídia.

Tabela 3.2 – Publicações selecionadas (cont.).

ID	Publicação	Descrição
40	(HÖLZL <i>et al.</i> , 2011)	Apresenta uma linguagem de padrões para apoiar os desenvolvedores de software a escolher ferramentas e técnicas apropriadas para desenvolver sistemas orientados a serviços com o apoio de métodos formais.
41	(BRAGA <i>et al.</i> , 2012)	Apresenta uma linguagem de padrões para estimativas de projetos de software ágeis. São oito padrões que, relacionados, ajudam equipes a obterem as estimativas necessárias para projetos de software ágeis.
42	(LYTRA <i>et al.</i> , 2012)	Apresenta uma linguagem de padrões para integração de plataformas baseadas em serviços cujos padrões foram definidos a partir de uma revisão sistemática de literatura. A LP apresentada é usada como base para a definição de um modelo de decisão arquitetural baseado em padrões.
43	(HAFIZ <i>et al.</i> , 2012)	Apresenta uma abordagem para desenvolvimento de uma linguagem de padrões para segurança a partir de um conjunto de padrões. A abordagem pode ser aplicada para desenvolver LPs para outros domínios relacionados a software.
44	(MUIJNCK-HUGHES; DUNCAN, 2012)	Apresenta uma linguagem de padrões para segurança de software que trata do projeto e implementação de sistemas de criptografia PBE (<i>Predicate Based Encryption</i>).
45	(ELORANTA; LEPPÄNEN, 2012)	Apresenta uma LP para interfaces gráficas para sistemas de controle de máquinas distribuídas. Cinco padrões da LP são discutidos em detalhes.
46	(ZIANI <i>et al.</i> , 2013)	Propõe uma abordagem de Engenharia Orientada a Modelos (<i>Model-Driven Engineering</i>) para produzir um repositório de artefatos de modelagem para apoiar o desenvolvimento de <i>Resource Constrained Embedded Systems</i> (RCES). Entre os artefatos estão padrões de <i>Security & Dependability</i> , que são armazenados em um repositório, permitindo que as relações entre eles sejam capturadas por ferramentas e representadas visualmente em uma linguagem de padrões.
47	(GUERRA <i>et al.</i> , 2013)	Apresenta uma linguagem de padrões para construção de <i>frameworks</i> baseados em metadados.
48	(FALBO <i>et al.</i> , 2013)	Apresenta SP-OPL (<i>Software Process Ontology Pattern Language</i>), uma linguagem de padrões ontológicos para apoiar o desenvolvimento de ontologias de processos de software.
49	(WANG <i>et al.</i> , 2013)	Apresenta uma linguagem de padrões que trata da conexão entre cenários e modelos de requisitos. Cada padrão conecta um aspecto relacionado a cenários a um modelo conceitual e oferece orientações sobre como converter o aspecto no referido modelo.
50	(MONTEIRO <i>et al.</i> , 2013)	A abordagem Galois para paralelização de algoritmos irregulares é tratada por meio de uma linguagem padrões para programação paralela que apresenta de forma geral os padrões de concorrência em algoritmos irregulares.
51	(FOGLI <i>et al.</i> , 2014)	Apresenta uma linguagem de padrões sobre acessibilidade, que pode ser utilizada para auxiliar <i>web designers</i> a criarem aplicações web acessíveis, de acordo com os mais recentes padrões.
52	(BARCELLOS <i>et al.</i> , 2014)	Apresenta M-OPL (<i>Measurement Ontology Pattern Language</i>) uma linguagem de padrões ontológicos para apoiar o desenvolvimento de ontologias de medição para diversos domínios.

Tabela 3.2 – Publicações selecionadas (cont.).

ID	Publicação	Descrição
53	(HENTRICH <i>et al.</i> , 2015)	Apresenta uma abordagem para suporte de mineração de padrões. Essa abordagem é aplicada em uma linguagem de padrões para apoiar arquitetos no projeto de arquiteturas orientadas a serviço. A linguagem contém padrões que abordam aspectos relacionados à modelagem, execução e integração de processos, visando auxiliar a definição de arquiteturas orientadas a serviço estáveis e passíveis de evolução.
54	(RUY <i>et al.</i> , 2015b)	Apresenta ISP-OPL (ISO- <i>Software Process Ontology Pattern Language</i>), uma linguagem de padrões ontológicos para processos de software definida considerando as normas ISO. Pode ser usada para a definição de ontologias para processos de software específicos e como base para harmonização dos padrões ISO relacionados a processos de software. Ela foi aplicada em um experimento com estudantes, que desenvolveram sete ontologias usando a LP.

QP1. Quando e em quais tipos de veículos (periódico/evento científico) as publicações foram publicadas?

Esta questão de pesquisa permite analisar quão amadurecida está a pesquisa no âmbito de linguagens de padrões de software. As publicações selecionadas foram publicadas nas últimas duas décadas. A Figura 3.2 mostra o número de publicações por ano.



Figura 3.2 - Relação de publicações por ano.

Em relação aos tipos de veículos, 67% dos estudos foram publicados em eventos científicos (sendo 61% em conferências e 6% em workshops, como mostra a Figura 3.3) e 33% foram publicados em periódicos.

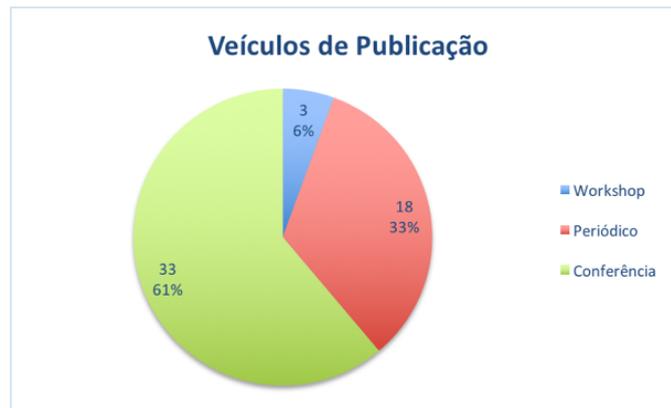


Figura 3.3 - Tipo de veículo de publicação dos estudos selecionados.

Levando-se em conta o período das publicações e seus veículos de publicação, pode-se notar maturidade no tópico de pesquisa, uma vez que vem sendo pesquisado há quase 20 anos e os resultados das pesquisas vêm sendo publicados em conferências e periódicos. O baixo percentual de publicações em workshops, que geralmente requerem menor maturidade nos trabalhos, pode ser entendido como um sinal de que as pesquisas nesse tópico estão amadurecidas o suficiente para serem publicadas em conferências e periódicos.

QP2. Quais tipos de pesquisa têm sido feitos?

Os estudos selecionados foram classificados de acordo com os tipos de pesquisa descritos em (WIERINGA *et al.*, 2006): Pesquisa de Avaliação, Proposta de Solução, Pesquisa de Validação, Artigo Filosófico, Artigo de Opinião, e Artigo de Experiência Pessoal. Estudos que apresentam LPs são Propostas de Solução. Estudos que aplicam LPs na prática, em casos reais, são Pesquisa de Avaliação. Estudos que aplicam LPs em exemplos são Pesquisa de Validação. A classificação dos estudos selecionados é apresentada na Figura 3.4. É possível observar que a grande maioria dos estudos (50) propõe LPs, sendo que 21 desses estudos realizaram algum tipo de avaliação. (FOGLI *et al.*, 2014) é um dos estudos que, além de propor uma LP, avalia sua aplicação em um estudo de caso. Outros estudos, tais como (MOURATIDIS *et al.*, 2005; SESERA, 2010; ZAMANI *et al.*, 2009; e ZIANI *et al.*, 2013), não propõem uma LP, porém, eles aplicam, em um caso real ou um exemplo, alguma LP preexistente.



Figura 3.4 - Tipos de pesquisa dos estudos selecionados.

QP3. As LPs propostas são de propósito geral ou específico?

As LPs apresentadas nos estudos foram definidas para auxiliar na produção de algum artefato relacionado a software. Nesse contexto, é possível observar dois níveis de generalidade em seu propósito, sendo algumas de propósito mais geral (independentes de domínio ou classe de aplicação), enquanto outras são para um domínio específico de aplicação ou de classe de sistema. A LP apresentada em (DELESSY *et al.*, 2007) é um exemplo de LP de propósito geral, pois é independente de domínio (pode ser aplicada em qualquer domínio) e de classe de aplicação (pode ser aplicada para qualquer tipo de sistema). Por outro lado, a LP apresentada em (FOGLI *et al.*, 2014), que trata aspectos relacionados à acessibilidade no desenvolvimento de sistemas web, tem propósito mais específico, pois embora seja independente de domínio, é voltada para uma classe de aplicação específica (sistemas web). A LP apresentada em (SESERA, 2010) para auxiliar no desenvolvimento de sistemas bancários também é de propósito específico, pois é para um domínio específico (bancário), assim como a LP do estudo (BRAGA; MASIERO, 2004), que é dependente de domínio (Gestão de Recursos Organizacionais).

Dentre as LPs encontradas nos estudos selecionados, 36% possuem um propósito mais geral, podendo ser utilizadas independentemente de domínio ou classe de sistema. Entretanto, a maioria das LPs, 64%, são direcionadas para um domínio específico de aplicação ou classe de sistema. A Figura 3.5 apresenta os dados relativos ao propósito das LPs.

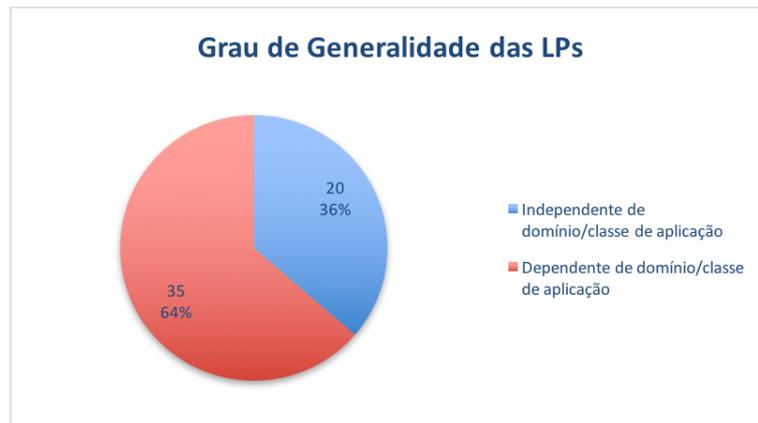


Figura 3.5 – Grau de Generalidade das LPs.

QP4. Quais tipos de padrões são encontrados nas linguagens de padrão propostas?

Essa questão de pesquisa visa analisar quais são os tipos de padrões organizados em LPs. Foram identificados diversos tipos, principalmente *Design Patterns*, Padrões Arquiteturais, Padrões de Processo, Padrões de Análise e Padrões Ontológicos. *Design Patterns* são adotados na Engenharia de Software para descrever soluções concretas para problemas recorrentes de projeto de software (FOGLI *et al.*, 2014). Padrões Arquiteturais expressam um esquema de organização estrutural fundamental para sistemas de software que especifica um conjunto de subsistemas, as suas responsabilidades e relacionamentos (SESERA, 2010). Padrões de Processo combinam técnicas gerais, ações e/ou tarefas para desenvolvimento de software, descrevendo de uma forma geral o que deve ser feito para solucionar um certo problema, sem se deter em detalhes (BRAGA *et al.*, 2012). Padrões de Análise se referem a padrões usados na fase de análise do desenvolvimento de software (SESERA, 2010). Por fim, Padrões Ontológicos, como informado na Seção 2.2, são soluções de modelagem usadas na resolução de problemas recorrentes no desenvolvimento de ontologias. Outros tipos de padrões foram encontrados nas LPs, porém, em menor ocorrência, tais como padrões de acesso, padrões instrucionais, padrões de interação, entre outros. A Figura 3.6 mostra a relação de LPs por tipo de padrão. Esses padrões de menor ocorrência foram agrupados na categoria “Outros”.



Figura 3.6 - Tipos de padrões encontrados nas LPs.

Conforme mostra a Figura 3.6, há uma predominância de padrões de projeto (*design patterns*), que são encontrados em aproximadamente 54% das LPs, e padrões arquiteturais, que são encontrados em 28% das LPs.

QP5. Quais elementos são tratados nas linguagens de padrões e que símbolos ou notações são utilizados para representá-los?

Em cada estudo selecionado há uma notação visual definida para a LP nele contida. Em alguns casos, os elementos da notação visual são definidos explicitamente. Por exemplo, em (BARCELLOS *et al.*, 2014) é apresentada uma legenda contendo os símbolos dos elementos usados na representação gráfica da linguagem e seu significado. Em outros casos, apenas é apresentada a representação gráfica da linguagem, sem uma indicação explícita do significado de seus elementos. Esse é o caso de (GUERRA *et al.*, 2013), por exemplo. Para responder QP5 quando não havia indicação explícita do significado dos seus elementos (em uma legenda, por exemplo), foi necessário identificar o significado dos elementos a partir da análise do conteúdo do artigo.

Algumas linguagens de padrões apresentam diversos elementos, como é o caso de (FALBO *et al.*, 2013), enquanto outras, como (BRAGA *et al.*, 2012; HSIEH *et al.*, 2011; KIRK *et al.*, 2007; WANG *et al.*, 2013 e WEYNS, 2009), limitam-se aos elementos *padrão* e *fluxo*, que são os elementos básicos para representar uma linguagem de padrões.

Além disso, embora várias LPs apresentem um mesmo elemento (por exemplo, todas as LPs analisadas apresentam o elemento *padrão*), são usados diferentes símbolos para representá-lo. Por exemplo, considerando todas as LPs analisadas, o elemento *padrão* é representado por sete símbolos diferentes, como mostra a Figura 3.7.

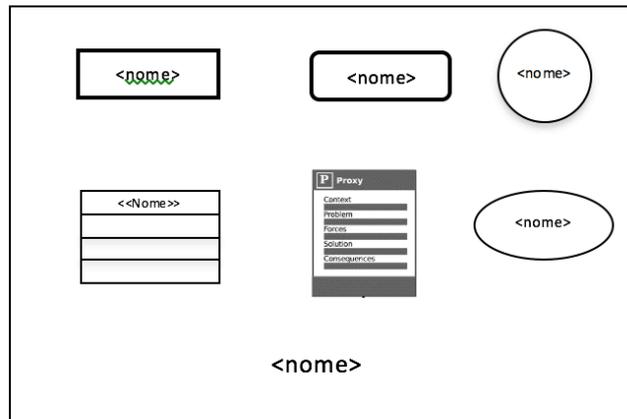


Figura 3.7 – Diferentes representações encontradas para o elemento padrão.

Na Figura 3.7, vale destacar ainda, a predominância de símbolos geométricos para representar padrões. Apesar de Moody (2009) considerar que elementos geométricos não demonstram, de forma direta, o significado do que representam, ou seja, são semanticamente opacos, esse tipo de representação é amplamente usado nos modelos das LPs e tradicionalmente usado na Engenharia de Software.

Os elementos encontrados nas LPs analisadas são:

- Padrão: elemento que trata de um problema recorrente.
- Grupo de Padrões: agrupamento de padrões.
- Subgrupo de Padrões: grupo contido em outro grupo.
- Fluxo: conexão entre elementos da LP que representa um caminho que pode ser seguido na LP.
- Fluxo Obrigatório: conexão entre elementos da LP que representa um caminho que deve obrigatoriamente ser seguido na LP.
- Fluxo Alternativo: conexão entre elementos da LP que representa um caminho que pode ser escolhido para ser seguido na LP como uma alternativa a outro caminho.
- Saídas Paralelas: conjunto de dois ou mais fluxos de saída definidos a partir de um fluxo de entrada.
- Entradas Paralelas: conjunto de dois ou mais fluxos de entrada que levam a um único fluxo de saída.
- Conector de Fluxo: nó usado para conectar fluxos.
- Ponto de Entrada: ponto que indica um padrão que pode ser o primeiro a ser aplicado.

- Relação Estrutural: relação que indica dependência entre os padrões.
- Relação Opcional: conexão entre elementos da LP que representa um caminho que pode (opcionalmente) ser seguido na LP.
- Padrões variantes: conjunto de dois ou mais padrões que resolvem o mesmo problema, mas de maneiras diferentes e mutuamente exclusivas.

A Figura 3.8 apresenta os elementos encontrados nas LPs investigadas. Na figura, as barras azuis indicam a quantidade de LPs que apresenta cada elemento. As barras vermelhas indicam quantas representações visuais diferentes foram encontradas para cada elemento.

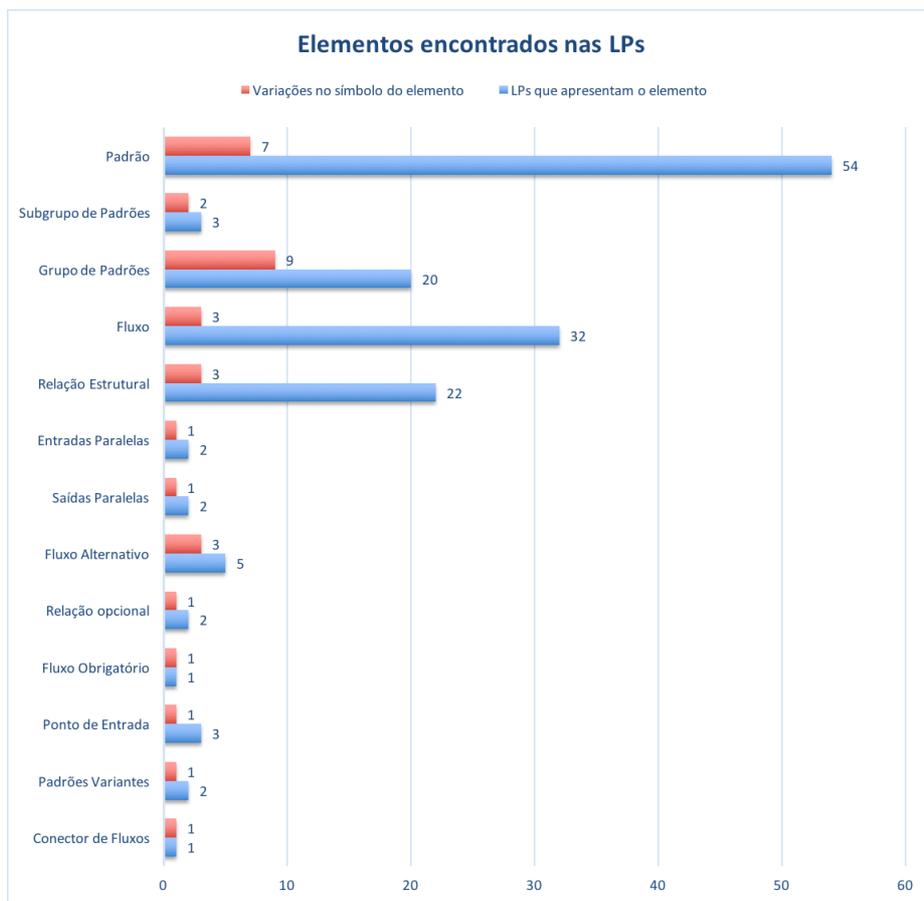


Figura 3.8: Elementos encontrados nas LPs e suas variações de representação visual.

QP6. As LPs definem os tipos de relações nela existentes? Se sim, quais os tipos de relações definidos?

Diferentes tipos de relações podem existir entre elementos de uma LP. Alguns estudos analisados não definem tipos diferentes de relações e limitam-se a indicar que há uma relação entre um elemento e outro (por exemplo, usando uma seta que conecta um padrão a outro). Exemplos desses estudos são (KIRK *et al.*, 2007; MUIJNCK-HUGHES; DUNCAN, 2012 e SESERA, 2010). Outros estudos indicam diferentes relações nomeando-

as informalmente com rótulos contendo nomes variados que não indicam tipos de relações e que são usados para facilitar a leitura do diagrama da LP (de maneira similar a nomes dados aos relacionamentos entre classes em um diagrama de classes UML). Exemplos desses estudos são (HENTRICH, 2009; SANTOS; KOSKIMIES, 2008 e WEYNS, 2009). 14 (26%) dos estudos selecionados encontram-se nas situações citadas, ou seja, não definem tipos específicos para as relações. 5 (9%) dos estudos definem apenas alguns tipos de relações entre os elementos, havendo no diagrama da LP relações com seus tipos definidos e outras não. Esse é o caso de (HENTRICH *et al.*, 2015) e (ZDUN, 2004b). Em 35 (65%) dos estudos os autores informam explicitamente os tipos das relações entre os padrões (dependência, alternativa, variância, entre outros). Exemplos de LPs onde os tipos de relações são explicitamente definidos são as apresentadas em (DELESSY *et al.*, 2007) e (ELORANTA; LEPPÄNEN, 2012). A Figura 3.9 apresenta os resultados obtidos para QP6.

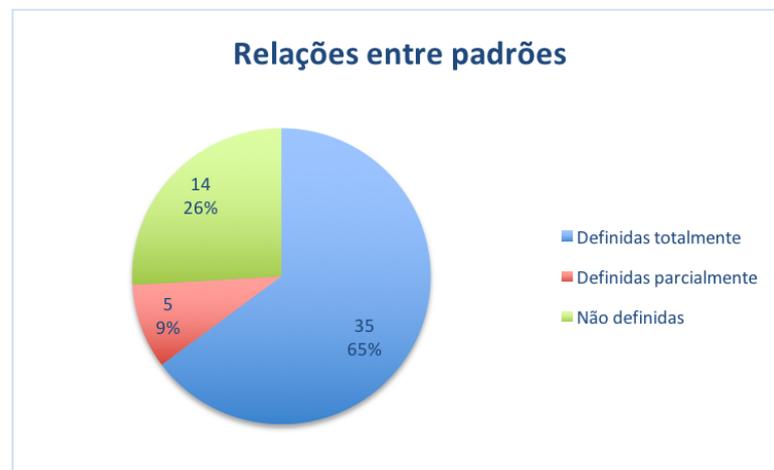


Figura 3.9 - Definição de tipos de relações nas LPs.

Na Figura 3.10 são apresentados os tipos de relações definidos nos estudos. A relação Fluxo é a que ocorre com maior predominância nas LPs analisadas. Algumas LPs apresentam apenas esse tipo de relação como, por exemplo, as LPs apresentadas em (BELLEBIA; DOUIN, 2006; MENG *et al.*, 2007 e WANG *et al.*, 2013). Outras apresentam mais de um tipo de relação, tais como as dos estudos (HAFIZ *et al.*, 2012; LYTRA *et al.*, 2012 e ZIANI *et al.*, 2013). Os tipos de relações identificados nas LPs analisadas são: *Fluxo*, que representa uma sequência de aplicação de padrões indicando que um padrão deve ser aplicado antes de outro; *Usa*, que indica que um padrão usa um outro padrão; *Dependência*, que representa uma relação estrutural de dependência entre padrões; *Especialização*, que indica que um padrão é a especialização de outro padrão, ou seja, soluciona um problema de forma mais específica que o padrão do qual é especializado; *Alternativa*, que indica padrões que podem ser usados de maneira alternativa; e *Variância*, que indica padrões mutuamente exclusivos e que resolvem

um mesmo problema, porém de formas diferentes. A classificação *Outras* foi utilizada para representar relações que apareceram com menor frequência nas LPs, tais como *implementa* e *realiza*, entre outras.

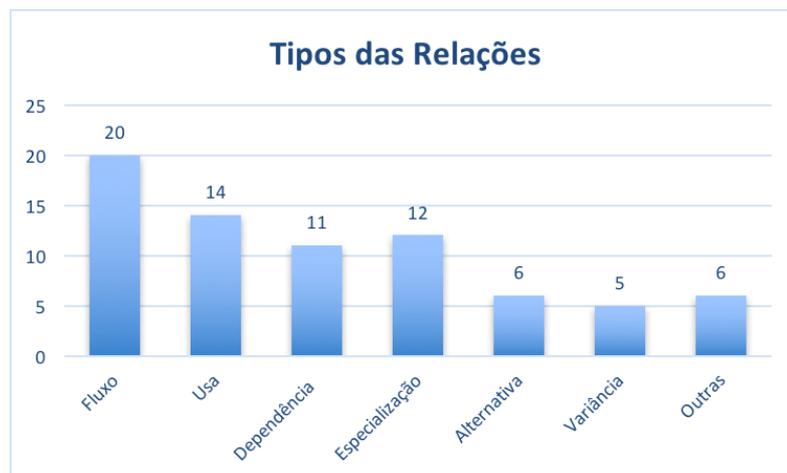


Figura 3.10 - Tipos de relações entre padrões.

QP7. Como padrões são agrupados nas LPs e como os agrupamentos são representados?

Agrupamentos foram encontrados nas LPs de 29 dos artigos selecionados. Na maioria deles (77%) padrões são agrupados de acordo com aspectos do domínio. Isso pode ser observado, por exemplo, em (BARCELLOS *et al.*, 2014) que apresenta uma linguagem de padrões ontológicos para medição de software e agrupa os padrões em seis grupos: Entidades Mensuráveis, Medidas, Unidades de Medidas & Escalas, Procedimentos de Medição, Planejamento de Medição e Medição & Análise. Em alguns estudos os padrões são agrupados considerando-se níveis de abstração. Esse é o caso de (FOGLI *et al.*, 2014) e (AVGERIOU; TANDLER, 2006). Outro aspecto considerado para agrupar padrões refere-se à natureza do problema. Isso pode ser observado, por exemplo, no estudo (BELLEBIA; DOUIN, 2006) que agrupou seus padrões em: Arquitetura, Configuração, Confiabilidade, Memória, Notificação, Topologia e Redes. Há, ainda, LPs que agrupam seus padrões de acordo com categorias (GZARA *et al.*, 2003), propósito do padrão (BRAGA; MASIERO, 2004) e fragmentos (NOBLE, 1998). A Figura 3.11 ilustra os resultados obtidos para esta QP. Vale destacar que uma mesma LP pode adotar mais de um critério para o agrupamento de padrões, como ocorre em (FOGLI *et al.*, 2014), que considera os critérios aspectos do domínio e níveis de abstração. Assim, a soma das ocorrências no gráfico da Figura 3.11 é superior a 29, que é o número de LPs que apresentam agrupamentos de padrões.

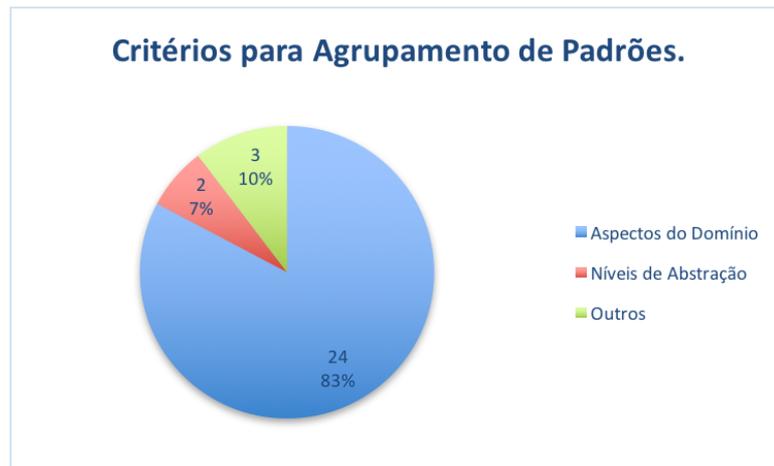


Figura 3.11: Critérios para agrupamentos de padrões nas LPs.

Os agrupamentos são representados de diversas formas nas LPs investigadas. O retângulo é o símbolo mais utilizado para indicar agrupamentos nas LPs. Estudos como (MEISTER *et al.*, 2004; SALAH; ZEID, 2009 e GUERRA *et al.*, 2013) usam essa representação. Outros estudos, tais como (HENTRICH, 2009; SESERA, 2010; MUIJNCK-HUGHES; DUNCAN, 2012 e HENTRICH *et al.*, 2015), utilizam um retângulo com cantos arredondados. A Figura 3.12 apresenta os diferentes símbolos utilizados para representar grupos de padrões nas LPs. Em seis dos 29 estudos considerados nesta questão de pesquisa, apesar de a publicação informar que os padrões da LP são organizados em grupos, eles não foram ilustrados na representação gráfica da LP provida na publicação. Por essa razão, esses estudos foram desconsiderados na extração de dados apresentados na Figura 3.12.

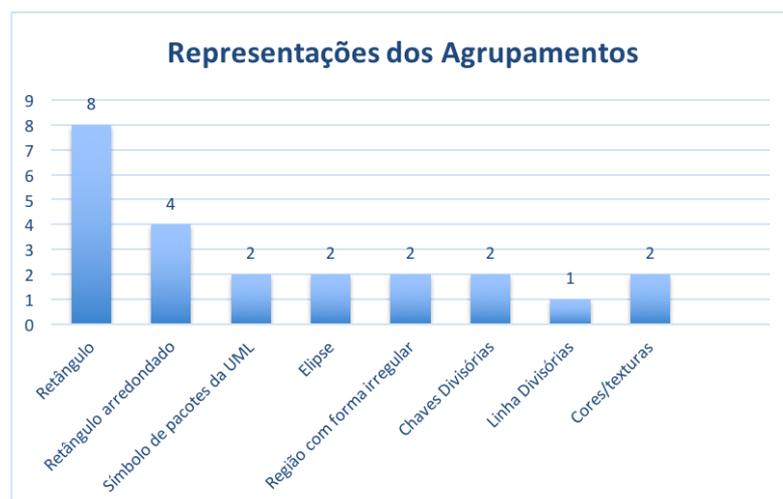


Figura 3.12 - Símbolos utilizados para representar grupos nas LPs.

QP8. Além da representação gráfica da linguagem, que mecanismo é fornecido para guiar a seleção dos padrões?

O propósito de uma linguagem de padrões é prover um guia para a seleção dos padrões. Dentre as publicações analisadas, 80% não proveem nenhum mecanismo além da representação gráfica da LP para guiar a seleção dos padrões. Em (RUY *et al.*, 2015b) é fornecida uma descrição do processo representado graficamente que guia a seleção dos padrões de acordo com os problemas a serem tratados. Essa descrição potencializa o entendimento da seleção e aplicação dos padrões, incrementando a eficácia da LP. Descrições do processo semelhantes à adotada em (RUY *et al.*, 2015b) são encontrados em outros estudos, tais como (BARCELLOS *et al.*, 2014), (FALBO *et al.*, 2013) e (GUERRA *et al.*, 2013). (ZDUN, 2004a) provê dois mecanismos para auxiliar na seleção dos padrões: uma descrição da LP com as possíveis seqüências de aplicação dos padrões e um modelos de características (*Feature Model*) para auxiliar na escolha da seqüência de padrões a serem aplicados. A Figura 3.13 apresenta os resultados obtidos para esta questão de pesquisa.

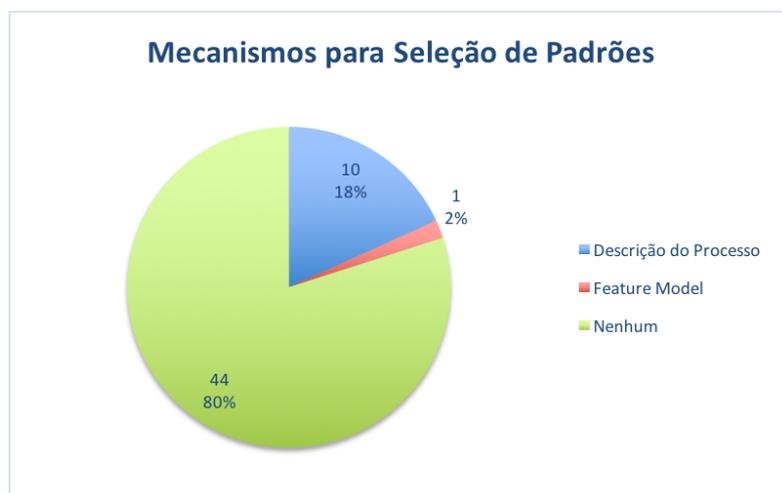


Figura 3.13 - Mecanismos utilizados pelas LPs para guiar a seleção dos padrões.

QP9. Quais são os tipos de modelos providos pela LP (estrutural / processo / ambos)?

LPs podem utilizar diferentes modelos para representar seus elementos, fornecendo diferentes visões que facilitam o entendimento e uso da linguagem. Modelos estruturais representam os elementos das LPs (por exemplo, padrões e grupos de padrões) e as relações entre eles. Modelos de processo, por sua vez, representam os caminhos possíveis para seleção e aplicação dos padrões. 46% dos estudos selecionados proveem modelos de processo indicando os fluxos possíveis de se seguir para a seleção dos padrões. Exemplos desses estudos são: (BRAGA *et al.*, 2012; HAFIZ *et al.*, 2012 e WANG *et al.*, 2013). Outros 46% dos estudos apresentam modelos que tratam dos relacionamentos de dependência estrutural entre os padrões. Como exemplo tem-se (NOBLE, 1998), que apresenta um modelo estrutural onde são utilizadas relações dos tipos “usa”, “especializa” e “realiza” para mostrar

as relações entre os padrões. Os 8% de estudos restantes (4 estudos) apresentam modelos para representar o processo de seleção dos padrões e também as relações existentes entre eles. Em 2 desses estudos, as LPs apresentam um modelo híbrido, ou seja, um único diagrama é usado para representar tanto relações de processo quanto relações estruturais entre padrões. Esse é o caso de (MOURATIDIS *et al.*, 2005) e (NOBLE, 1998). Nos outros dois, (ZDUN, 2004a) e (GUERRA *et al.*, 2013), são apresentados dois modelos, um para guiar o processo de aplicação dos padrões da LP e outro para mostrar as relações estruturais entre eles. A Figura 3.14 apresenta esses resultados.

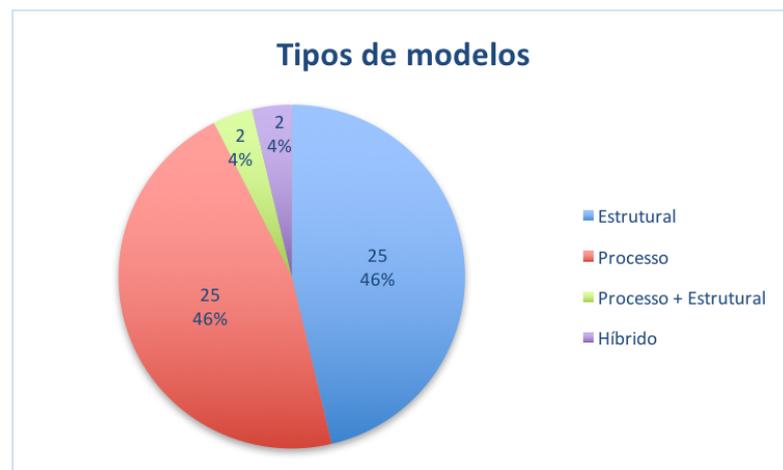


Figura 3.14: Tipo dos modelos apresentados nos estudos.

3.5 Discussões

Nesta seção são realizadas discussões e são fornecidas algumas informações adicionais relativas aos resultados apresentados na seção anterior.

No estudo realizado, foram analisadas mais de 50 LPs desenvolvidas com diversos propósitos, entre eles: apoiar o desenvolvimento de sistemas, *frameworks* ou plataformas, prover soluções para problemas de domínio e auxiliar no desenvolvimento de projetos, modelos ou ontologias. As LPs apresentadas nas publicações analisadas estão relacionadas a diversas subáreas da Engenharia de Software. No âmbito de Processo de Software, há LPs relacionadas a definição de processos de software (MENG *et al.*, 2007; FALBO *et al.*, 2013 e RUY *et al.*, 2015b), medição de software (BARCELLOS *et al.*, 2014), gerência de projetos, engenharia de requisitos (WANG *et al.*, 2013) e projeto de sistemas (KHALED; HOSNY, 2006) e (AHLUWALIA, 2007). Há, também, propostas direcionadas à Arquitetura de Software, que visam auxiliar o desenvolvimento de arquiteturas de software para *frameworks* (KIRK *et al.*, 2007) e (GUERRA *et al.*, 2013), integração de plataformas (LYTRA *et al.*, 2012), Sistemas Multiagentes (WEYNS, 2009), Sistemas de Controle de Máquinas (ELORANTA *et*

al., 2009) e Sistemas *Peer-to-peer* (GROLIMUND; MÜLLER, 2006). Além disso, fornecem soluções para problemas de arquiteturas orientadas a serviços ou processos (HENTRICH; ZDUN, 2007; HENTRICH, 2009 e HENTRICH *et al.*, 2015), arquiteturas baseadas em aspectos de usabilidade (JOHN *et al.*, 2009) e organização de padrões arquiteturais (AVGERIOU; ZDUN, 2005). No contexto de desenvolvimento de software, há LPs relacionadas à segurança de software (MAHEMOFF *et al.*, 2001; MOURATIDIS *et al.*, 2005; DELESSY *et al.*, 2007; HAFIZ *et al.*, 2012 e MUIJNCK-HUGHES; DUNCAN, 2012), interface com o usuário (BORCHERS, 2001; WELLHAUSEN, 2005 e ELORANTA; LEPPÄNEN, 2012), acessibilidade web (FOGLI *et al.*, 2014) e programação (NOBLE, 1998b; GOEDICKE *et al.*, 2004; ZDUN, 2004a, 2004b; MONTEIRO; AGUIAR, 2007 e MONTEIRO *et al.*, 2013). Há, ainda, LPs que abordam softwares gratuitos, livres e *open source* (HANNEBAUER *et al.*, 2010) e outras destinadas a domínios de aplicação bem específicos, como, por exemplo, sistemas de Telefonia (HANMER; KOCAN, 2004)

Ao se analisar os tipos de padrões encontrados nas LPs, notou-se predominância dos *design patterns*. Isso pode ser entendido como um sinal de que as pesquisas relacionadas a esse tipo de padrão estão mais amadurecidas que as pesquisas relacionadas a LPs que envolvem outros tipos de padrões. Foram encontradas apenas três LPs com padrões ontológicos (FALBO *et al.*, 2013; BARCELLOS *et al.*, 2014 e RUY *et al.*, 2015b) sendo todas bem recentes e desenvolvidas pelo mesmo grupo de pesquisa.

Quanto aos elementos tratados nas LPs, há diversidade de elementos e também de símbolos utilizados para representá-los. Também notou-se heterogeneidade na quantidade de elementos considerados em cada LP. Em (BARCELLOS *et al.*, 2014), por exemplo, são considerados sete elementos: ponto de entrada, padrão, fluxo, variância, nó de decisão, grupo e subgrupo. Já em (ZDUN, 2004a), apenas dois elementos são tratados: padrão e relacionamento de dependência. A variação na quantidade de elementos tratados nas LPs pode indicar que: (a) as LPs tratam situações diferentes e, com isso, em algumas são necessários mais elementos do que em outras, ou (b) algumas LPs utilizam menos elementos do que o necessário para que a LP seja entendida adequadamente. Nesse caso, ocorre sobrecarga no significado de elementos e dos símbolos que os representam, o que leva a LPs pouco claras e cognitivamente pobres. De fato, em vários estudos, diversas dificuldades foram encontradas para entender as linguagens devido à falta de clareza de sua representação visual.

Em relação aos tipos de relações, notou-se que nem sempre as LPs definem explicitamente as diferentes relações possíveis entre os elementos, o que dificulta o

entendimento das relações existentes entre os padrões e, conseqüentemente, sua seleção e aplicação. Dentre os estudos que definem tipos de relações, há predominância da relação de dependência, seguida por uso e especialização. De fato, não surpreende haver predominância da relação dependência nas LPs, uma vez que ela pode englobar outras e eliminar (ou mascarar) a necessidade de definir outros tipos. Por exemplo, se um padrão A usa um padrão B, pode-se identificar uma relação de uso de A para B (A usa B) ou, de forma mais geral, pode-se entender que há uma relação de dependência entre A e B (A depende de B). No entanto, ao não especificar que a relação entre A e B é de uso, não é possível identificar exatamente qual é a relação entre A e B. Por exemplo, A depende de B, pois só pode ser aplicado depois de B, uma vez que requer a solução dada por B, ou A depende de B, pois B é parte da solução proposta em A? Em ambos os casos pode-se dizer que A depende de B, mas no segundo caso, essa dependência é especificamente uma relação de uso. Alguns tipos de relação, como complemento e implementação, são pouco utilizados nas linguagens. Em alguns casos, esses tipos de relação podem ser definidos a partir de outros. Por exemplo, pode-se considerar que a relação de complemento pode ser definida a partir da relação de uso. Em um modelo estrutural, se um padrão A complementa um padrão B, pode-se entender que o padrão B usa A.

Várias das LPs analisadas agrupam padrões. A maioria define os agrupamentos considerando aspectos do domínio. Se por um lado há uma predominância nos critérios usados para agrupar padrões (77% dos trabalhos agrupam padrões considerando aspectos do domínio), por outro destaca-se a grande variedade de símbolos utilizados para representar os grupos nas LPs. Uma característica comum à maioria deles é o uso de símbolos que delimitam uma região na qual os padrões são agrupados. A definição de grupos é particularmente importante em LPs grandes e complexas. O agrupamento de padrões pode ser uma estratégia para auxiliar o gerenciamento da complexidade da LP. Grupos podem ser representados de maneira transparente (i.e., os padrões presentes em cada grupo são visíveis na LP) ou como caixas pretas (i.e., a LP apresenta apenas o símbolo que representa o grupo, sem apresentar os padrões que fazem parte dele). Dessa forma, pode-se fornecer visões da LP com diferentes níveis de detalhes, o que pode contribuir para o entendimento da LP. A LP de (HAFIZ *et al.*, 2012) é a única que apresenta uma representação caixa-preta dos grupos de padrões. Para minimizar a complexidade do modelo, algumas partes da LP são apresentadas parcialmente, omitindo as relações internas existentes nos grupo de padrões (representados pelo nome do grupo em negrito). Dessa forma, a LP tem duas possíveis visões: uma visão geral da LP, apresentando todos os padrões e suas relações, e uma visão

específica, que ilustra apenas pequenas partes da LP e não mostra as relações internas dos grupos destacados pelos nomes em negrito.

A maioria das LPs analisadas não fornece, além dos modelos da LP, mecanismos de apoio à seleção dos padrões. Apenas 10 delas fornecem uma descrição de sua representação visual e, dessas, apenas algumas apresentam essa descrição de maneira clara e detalhada, capaz de servir como um guia para a seleção dos padrões. Em (ZDUN, 2004a; ZAMANI *et al.*, 2009; FALBO *et al.*, 2013; BARCELLOS *et al.*, 2014 e RUY *et al.*, 2015b) são apresentadas descrições detalhadas, que contêm orientações que informam diante de qual situação um ou outro padrão deve ser selecionado.

Por fim, em relação aos tipos de modelos utilizados para representar as linguagens, notou-se que em 92% dos estudos apenas um tipo de modelo é apresentado, provendo uma visão ou do processo ou da estrutura da LP. Poucos são os estudos que apresentam as duas visões e, dentre esses, alguns utilizam um mesmo diagrama para isso, abordagem que tende a dificultar o entendimento do modelo e a percepção das diferentes visões. A existência de um modelo apresentando os possíveis caminhos para seleção e aplicação dos padrões e de outro modelo mostrando as relações entre os elementos da linguagem pode contribuir para um melhor entendimento e uso da LP.

Considerando-se o panorama provido a partir dos resultados deste mapeamento sistemático, as seguintes questões relacionadas ao tópico de pesquisa investigado podem ser destacadas: (i) não há padrão para as notações visuais das LPs, havendo, inclusive, inconsistências nas notações usadas em trabalhos de um mesmo grupo de pesquisa (por exemplo, em (BARCELLOS *et al.*, 2014; FALBO *et al.*, 2013 e RUY *et al.*, 2015b); (ii) as notações visuais adotadas são cognitivamente pobres; (iii) não há preocupação em prover mecanismos para auxiliar a seleção dos padrões; e (iv) ausência de uma visão completa das LPs, que apresente aspectos relacionados ao processo da LP e à sua estrutura.

3.6 Considerações finais

Este capítulo apresentou os resultados de um mapeamento sistemático que investigou notações visuais adotadas em Linguagens de Padrões de Software. Os resultados obtidos mostram que há várias questões relacionadas às notações visuais de LPs que ainda precisam ser exploradas.

No Capítulo 2 foram apresentados alguns princípios que devem ser considerados para se obter notações visuais cognitivamente ricas. Os princípios de PoN (MOODY, 2009) podem ser utilizados na avaliação da qualidade de notações visuais, bem como em seu *design*.

Analisando-se as notações das LPs investigadas no mapeamento sistemático, pode-se observar que vários dos princípios sugeridos por Moody (2009) não são atendidos. Em algumas LPs nota-se sobrecarga de símbolos e, em outras, déficit de símbolos. Em ambos os casos, o princípio de clareza semiótica não é atendido. Além disso, devido à pequena distância visual encontrada entre os símbolos na maioria das LPs e ao pequeno número de variáveis visuais exploradas nas notações, o atendimento aos princípios discriminabilidade perceptiva e expressividade visual também fica comprometido.

Apesar do não atendimento a vários princípios de PoN, é possível observar aderência a alguns deles, tais como economia gráfica e codificação dupla. Além disso, apesar de não haver uma grande distância visual entre os símbolos, nota-se predominância do uso de formas geométrica para modelar linguagens de padrões, simbolismo comum na área de modelagem, que potencializa o aprendizado das linguagens e contribui para o atendimento ao princípio de transparência semiótica.

Para que LPs possam ser adequadamente utilizadas e possam prover a seus usuários os benefícios esperados, é importante que seja utilizada uma notação visual de qualidade, cognitivamente rica, que proveja visões de processo e estrutural e forneça mecanismos que guiem na seleção dos padrões. Essas questões foram levadas em consideração no desenvolvimento da notação visual proposta neste trabalho.

Essas questões emergiram da investigação de notações visuais de LPs de software. Para obter informações referentes a notações visuais adotadas para representar linguagens de padrões ontológicos (LPOs), decidiu-se por avaliar a notação utilizada para representar uma das LPOs identificadas no mapeamento, a qual foi desenvolvida no NEMO (mesmo grupo de pesquisa em que este trabalho foi realizado). Para isso, foi realizado um estudo experimental, o qual é apresentado no próximo capítulo.

Capítulo 4

Estudo Experimental para Avaliação de uma Linguagem de Padrões Ontológicos

Este capítulo apresenta um estudo experimental realizado para avaliar a utilização de LPOs no desenvolvimento de ontologias e a notação visual até então utilizada para representar uma LPO. Na Seção 4.1 é apresentada uma breve descrição da LPO utilizada no estudo. Na Seção 4.2 é apresentado o planejamento do estudo. Na Seção 4.3 são apresentados os resultados obtidos. Na Seção 4.4 algumas discussões sobre os resultados são realizadas. A Seção 4.5 aborda ameaças à validade do estudo. Por fim, na Seção 4.6 são feitas as considerações finais deste capítulo.

4.1 ISO-based Software Process Ontology Pattern Language (ISP – OPL)

Para realizar o estudo, foi utilizada a versão inicial de *ISO-based Software Process Ontology Pattern Language* (ISP-OPL) (RUY *et al.*, 2015a) uma LPO que apoia a construção de ontologias no domínio de processo de software, tendo as normas ISO como a principal fonte de conhecimento.

ISP-OPL aborda os seguintes aspectos de processo de software: Unidades de Trabalho (*Work Units – WU*), Recursos Humanos (*Human Resources – HR*) e Produtos de Trabalho (*Work Products – WP*).

A Figura 4.1 mostra o modelo de processo da ISP-OPL, representado na forma de um diagrama de atividades da UML estendido. No modelo, padrões ontológicos relacionados a um domínio (*Domain-Related Ontology Patterns - DROPs*) são representados pelos nós de ação (retângulos arredondados com *labels*); Nós Iniciais (círculos sólidos) representam pontos de entrada na LPO, ou seja, pontos que indicam os padrões da LPO que podem ser usados primeiro, sem que seja necessário solucionar outros problemas previamente; Fluxos de controle (setas) representam as sequências admissíveis de uso dos padrões; Nós de fusão (*merge*) (símbolo no formato de diamante) representam os pontos onde há uma fusão de caminhos na LPO; Nós de Junção/Bifurcação (barra sólida) representam a conjunção de caminhos (junção) ou caminhos paralelos possíveis (bifurcação); Uma extensão da notação original da UML (linhas pontilhadas com setas) é usada para representar padrões variantes, ou seja, padrões que podem ser usados para solucionar um mesmo problema de diferentes maneiras e que são mutuamente exclusivos. Por fim, padrões são agrupados de acordo com os aspectos do processo de software aos quais se relacionam. O símbolo utilizado para representar esses agrupamentos é um retângulo com bordas azuis.

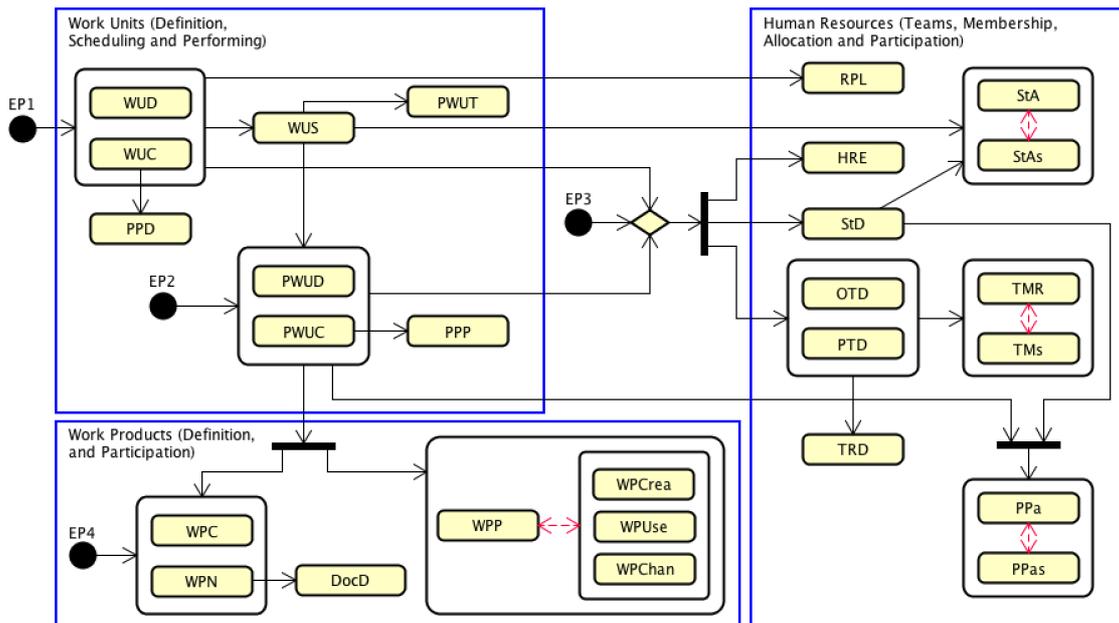


Figura 4.1 - Modelo de Processo da ISP-OPL original (RUY *et al.*, 2015a).

Como mostra a Figura 4.1, ISP-OPL tem quatro pontos de entrada. O engenheiro de ontologias deve escolher um desses pontos, de acordo com o escopo da ontologia de processo de software específica que será desenvolvida. EP1 deve ser escolhido quando os requisitos para a ontologia a ser desenvolvida incluem a definição e planejamento de unidades de trabalho (WU). Porém, se o escopo da ontologia considera apenas a execução das unidades de trabalho, o engenheiro de ontologias deve escolher EP2. Caso o escopo da ontologia aborde apenas assuntos relacionados a Recursos Humanos, então EP3 deve ser selecionado. Finalmente, EP4 deve ser escolhido se o engenheiro de ontologias necessita modelar apenas a estrutura dos produtos de trabalho. A partir desses pontos de entrada, o engenheiro de ontologias deve seguir os caminhos indicados pelos fluxos e selecionar os padrões de acordo com os problemas por eles tratados, i.e., se tratar o problema considerado por um padrão faz parte do escopo da ontologia, então o padrão deve ser selecionado, obedecendo-se as sequências indicadas pelos fluxos. Vale destacar que no modelo de processo de ISP-OPL, não há um nó final indicando o término do processo. Dessa forma, o engenheiro de ontologias pode optar por parar a execução do processo assim que considerar que seu escopo foi atendido.

A documentação completa de ISP-OPL usada no estudo pode ser encontrada em <http://www.inf.ufes.br/~falbo/files/ISP-OPL%20Specification.v0.6.vf.pdf>.

4.2 Planejamento do Estudo

Tratados no âmbito da Engenharia de Software Experimental, os estudos experimentais têm sido usados para encontrar indícios e aprimorar a utilização de técnicas no contexto de projetos de software (TRAVASSOS *et al.*, 2002). No contexto deste trabalho foi realizado um *survey* buscando-se encontrar indícios que permitam avaliar e aprimorar a notação visual utilizada até então na representação de LPOs. Um *survey* é um estudo experimental realizado em retrospecto quando, por exemplo, uma nova tecnologia ou método é utilizado e deseja-se obter informações que permitam avaliá-los e evoluí-los (PFLEEGER, 1994).

O estudo foi realizado com alunos de mestrado e doutorado do Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo e ocorreu no contexto da disciplina Ontologias para Engenharia de Software ministrada no segundo semestre de 2014. Na disciplina são abordados os seguintes tópicos: Ontologias - Definições e Tipos; Ontologias e Engenharia de Software; Ontologias do Domínio de Engenharia de Software; Aplicações de Ontologias em Engenharia de Software - Interoperabilidade Semântica, Ontologias no Desenvolvimento de Software, Semântica em Aplicações de Software.

O **objetivo** do estudo realizado foi avaliar o uso de uma LPO no desenvolvimento de ontologias de domínio e a notação visual adotada. Utilizando-se a abordagem GQM (BASILI *et al.*, 1994) este objetivo é assim formalizado:

Analisar *uma Linguagem de Padrões Ontológicos*

Com o propósito de *avaliar (i) sua aplicação e (ii) a notação visual utilizada para representá-la*

Com respeito *(i) aos impactos do uso da LPO no desenvolvimento de ontologias e (ii) à facilidade de a notação ser entendida e à capacidade de representar adequadamente uma LPO*

Sob o ponto de vista de *engenheiros de ontologias*

No contexto de *engenharia de ontologias de domínio*

Para analisar os resultados, foram considerados os seguintes indicadores:

- a) Produtividade no desenvolvimento de ontologias
- b) Qualidade da ontologia resultante
- c) Grau de dificuldade para entendimento do mecanismo de funcionamento da LPO
- d) Grau de dificuldade para entendimento dos símbolos da LPO
- e) Ausência de Problemas de Representação da Notação

A **instrumentação** utilizada na condução do estudo consiste de três formulários: (i) um termo de consentimento para a realização do estudo, que visa resguardar os direitos dos participantes quanto ao estudo e seus resultados; (ii) um formulário para caracterizar o perfil dos participantes, que visa obter informações sobre o conhecimento e experiência dos participantes em modelagem conceitual, desenvolvimento de ontologias e LPOs; e (iii) um questionário que permite que os participantes registrem sua percepção após o uso da LPO para apoiar o desenvolvimento de ontologias para subdomínios da Engenharia de Software. Esses formulários são apresentados no Apêndice A desta dissertação.

O **procedimento** de condução do estudo consistiu de três etapas. Na primeira etapa, pesquisadores explicaram o contexto do estudo aos participantes e apresentaram ISP-OPL. A documentação necessária para uso da LPO foi disponibilizada para os participantes, compreendendo uma listagem com a descrição resumida de cada padrão ontológico presentes na LPO, o diagrama de processo e sua descrição, e a descrição detalhada de cada padrão, contendo nome, intenção, *rationale*, questões de competência, modelo conceitual e axiomatização. Na segunda etapa, os participantes do estudo organizaram-se em grupos e utilizaram ISP-OPL para desenvolver ontologias de domínio relacionadas a processos de software, tendo sido produzidas ontologias para os seguintes processos: Gerenciamento de Recursos Humanos, Medição, Gerenciamento de Riscos, Projeto Arquitetural de Software, Gerenciamento de Configuração de Software, Gerenciamento de Documentação de Software e Manutenção de Software. Durante essa etapa não houve intervenção por parte dos pesquisadores. Os participantes desenvolveram as ontologias como um trabalho da disciplina para avaliação e atribuição de nota, 2 meses após a apresentação de ISP-OPL e disponibilização da documentação. Para desenvolver as ontologias os participantes utilizaram modelos de maturidade em processos de software e normas ISO como fontes de conhecimento sobre o domínio a ser tratado. A terceira etapa consistiu na aplicação de um questionário para registrar a percepção dos participantes do estudo sobre o uso da LPO e a notação utilizada para representar seu processo. Após receber as respostas ao questionário, foram realizadas entrevistas a fim de se obter informações mais detalhadas sobre a experiência de uso da LPO.

O **questionário** aplicado inclui questões relacionadas à representação da LPO e também questões relacionadas ao uso da LPO no desenvolvimento de ontologias de domínio. O questionário contém questões objetivas e discursivas. Para as objetivas é

solicitado ao participante que incluía uma justificativa. Algumas questões do questionário são apresentadas na Figura 4.2. O questionário pode ser visto na íntegra no Apêndice A.

 UFES (Universidade Federal do Espírito Santo) NEMO (Ontology & Conceptual Modeling Research Group) Estudo empírico realizado na disciplina Ontologias para Engenharia de Software Experimento: Análise do uso de uma OPL para criação de uma ontologia de domínio.	
Questionário de Avaliação de Atividade	
Nome:	
Responda as questões abaixo considerando sua percepção inicial da OPL.	
01	Quão difícil/fácil você considera entender o mecanismo geral de funcionamento da OPL, que é composto por um processo e um conjunto de <i>patterns</i> ?
<input type="checkbox"/> Muito difícil <input type="checkbox"/> Difícil <input type="checkbox"/> Neutro <input type="checkbox"/> Fácil <input type="checkbox"/> Muito Fácil	
Justificativa:	
02	Quão difícil/fácil você considera entender cada uma das representações dos elementos da OPL:
● Entry Point	<input type="checkbox"/> Muito difícil <input type="checkbox"/> Difícil <input type="checkbox"/> Neutro <input type="checkbox"/> Fácil <input type="checkbox"/> Muito Fácil
Justificativa:	
 Pattern	<input type="checkbox"/> Muito difícil <input type="checkbox"/> Difícil <input type="checkbox"/> Neutro <input type="checkbox"/> Fácil <input type="checkbox"/> Muito Fácil
Justificativa:	
→ Path	<input type="checkbox"/> Muito difícil <input type="checkbox"/> Difícil <input type="checkbox"/> Neutro <input type="checkbox"/> Fácil <input type="checkbox"/> Muito Fácil

Figura 4.2 - Fragmento do Questionário de Avaliação

Os **participantes** do estudo foram 19 estudantes do Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, com conhecimento em modelagem conceitual. Desses, 2 eram alunos de doutorado e 17 de mestrado. Dentre os participantes, 4 declararam ter experiência *alta* em modelagem conceitual, 9 declararam ter experiência *média* e 4 declararam ter *baixa* experiência. Em relação à experiência em desenvolvimento de ontologias, 3 participantes declararam ter *média* experiência, 7 declararam ter experiência *baixa* e 8 declararam ser sua primeira experiência no desenvolvimento de ontologias. Por fim, em relação à experiência em LPOs, todos declararam que o estudo foi o primeiro contato com LPOs.

Embora a maioria dos participantes do estudo tenha pouca ou nenhuma experiência com desenvolvimento de ontologias e o uso de uma LPO tenha sido inédito para todos, eles não foram excluídos do estudo, uma vez que todos possuem conhecimento em modelagem conceitual e representam perfis possíveis de usuários de LPOs, já que usuários de LPOs podem ser desde engenheiros de ontologias experientes até iniciantes. De toda forma, a avaliação dos resultados deve levar em consideração o perfil dos participantes do estudo.

4.3 Resultados

Nesta seção são apresentados os principais resultados obtidos a partir dos questionários respondidos pelos participantes do estudo.

4.3.1 Percepção acerca da LPO

As primeiras perguntas feitas para os participantes do estudo dizem respeito à representação da LPO e visam capturar a percepção dos participantes quanto ao entendimento do funcionamento da LPO e dos símbolos utilizados.

Em relação ao mecanismo geral de funcionamento da LPO, que é composto por um processo e um conjunto de padrões, a maioria dos participantes considerou *fácil* (8 participantes) ou *muito fácil* (2 participantes) entendê-lo, enquanto 4 acharam *difícil* e 5 responderam *neutro*. As respostas estão ilustradas na Figura 4.3.

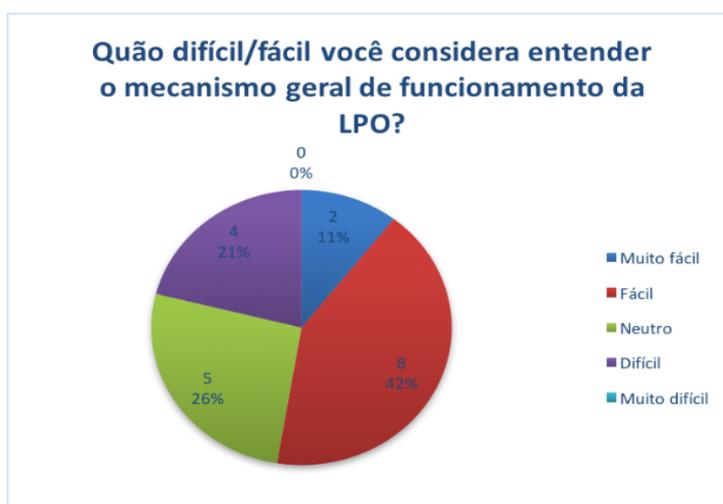


Figura 4.3 – Compreensão do funcionamento da OPL.

Com relação aos símbolos utilizados para representar o processo da LPO, no geral, houve uma boa aceitação por parte da maioria dos participantes. 3 deles declararam ter tido dificuldades no entendimento do símbolo de padrões variantes (*Variant Patterns*), 2 com os símbolos dos nós de junção (*Join Node*) e bifurcação (*Fork Node*) e 1 com o símbolo usado para representar padrões (*Patterns*). A Figura 4.4 apresenta essas respostas mais detalhadamente.

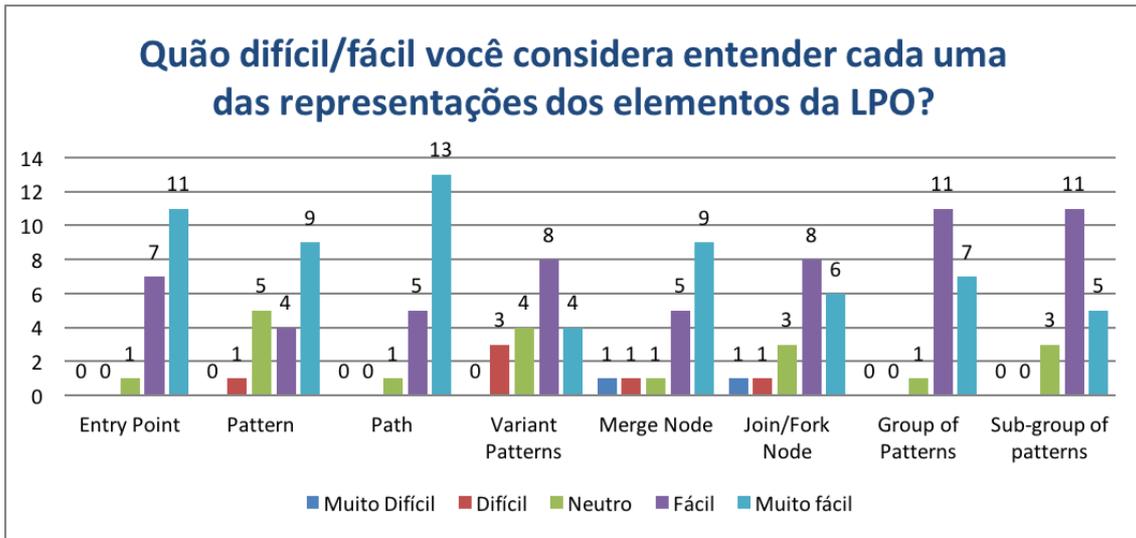


Figura 4.4 – Compreensão dos símbolos dos elementos da LPO.

No que se refere ao símbolo que os participantes consideraram mais difícil para entender, a representação adotada para padrões variantes (*Variant Patterns*) foi a mais citada pelos participantes, que justificaram não entender a característica exclusiva do elemento representado (uso exclusivo de um dos padrões). Nos casos dos nós de fusão, junção e bifurcação (*Merge Node*, *Join Node* e *Fork Node*, respectivamente), as justificativas foram voltadas à falta de lembrança do significado do símbolo na UML e a não ficar claro ser obrigatório seguir os vários caminhos ou apenas um deles. As dificuldades relatadas para entendimento do símbolo adotado para representar padrões (*Pattern*) dizem respeito ao uso de siglas para nomeá-los. Um participante enfatizou que as siglas não auxiliavam a entender o significado do padrão. A Figura 4.5 ilustra os resultados obtidos.

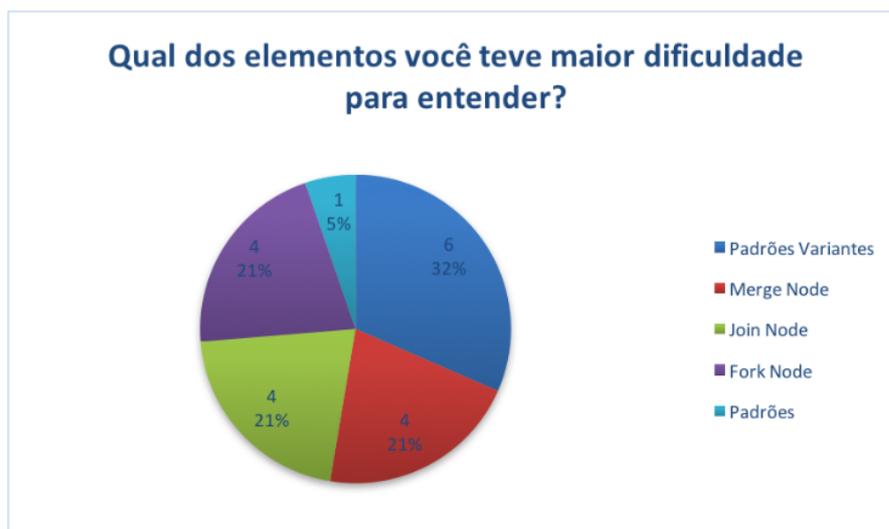


Figura 4.5 - Elementos mais difíceis de entender.

Foi solicitada aos participantes a identificação de possíveis melhorias na representação da LPO. Dentre as sugestões de melhorias dadas pelos participantes, as que mais se destacaram foram: deixar mais claro no símbolo de padrões variantes que há exclusividade de aplicação entre os padrões, identificar caminhos opcionais e obrigatórios, e nomear os padrões de forma a sugerir o seu significado.

4.3.2 Percepção acerca do uso de LPOs no desenvolvimento de ontologias de domínio

Após as perguntas sobre a representação de LPOs, foram realizadas perguntas sobre o uso da LPO no desenvolvimento de uma ontologia de domínio, a fim de capturar a percepção dos participantes quanto à utilização da LPO e os impactos dessa abordagem na produtividade no desenvolvimento e na qualidade da ontologia resultante.

Em relação ao grau de dificuldade encontrado no desenvolvimento da ontologia de domínio com o uso da LPO, a maioria dos participantes considerou *fácil* (11 participantes) ou *muito fácil* (1 participante) criar uma ontologia a partir de uma LPO. 5 informaram *neutro* e 2 consideraram o processo *difícil*. A Figura 4.6 apresenta esses resultados.

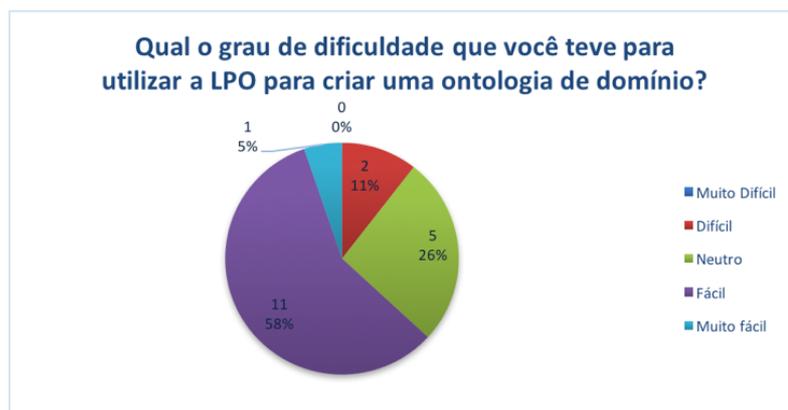


Figura 4.6 - Grau de dificuldade no uso de LPOs na criação de ontologias de domínio.

Com relação à produtividade no processo de desenvolvimento de ontologias de domínio com LPOs, apenas os participantes que já tinham alguma experiência na criação de ontologias responderam a essa questão (10 participantes), uma vez que os demais não poderiam avaliar a diferença de desenvolver uma ontologia de domínio a partir do nada ou usando uma LPO. Todos os participantes avaliaram positivamente o uso da LPO, sendo que muitos consideraram o desenvolvimento da ontologia através da LPO um processo mais fácil e rápido quando comparado a não usar uma LPO.

Em relação à qualidade da ontologia resultante do uso de uma LPO, as respostas foram positivas. Todos os participantes consideraram que o uso de LPOs contribuiu para a

qualidade da ontologia resultante. Alguns justificaram que a LPO ajudou a identificar alguns conceitos que deveriam aparecer na ontologia, de modo que ela ficou completa e bem fundamentada.

Dentre as dificuldades encontradas no uso da LPO, alguns participantes destacaram a identificação de conceitos específicos da ontologia de domínio.

4.3.3 Entrevistas

Após a aplicação do questionário, foram realizadas entrevistas com os grupos de participantes do experimento com a finalidade de obter informações adicionais e esclarecimentos sobre comentários registrados nos questionários respondidos. As perguntas realizadas nas entrevistas são apresentadas no Apêndice A desta dissertação.

Em relação ao desenvolvimento de ontologias de domínio utilizando LPOs, a maioria dos grupos entrevistados declarou que foi um processo produtivo e de fácil condução. Segundo os participantes, a partir dos padrões da LPOs, os quais são bem fundamentados, foi possível obter um aprofundamento do entendimento do domínio em paralelo com a criação da ontologia. Isso influenciou diretamente no tempo de desenvolvimento e na qualidade da ontologia de domínio. Outro ponto destacado pelos participantes foi o fato de os padrões e seus relacionamentos descritos na LPO terem auxiliado na definição do escopo da ontologia de domínio, tanto na disposição de informações não obtidas previamente por eles, quanto na limitação do nível de abstração do domínio.

No que se refere à exploração de conceitos específicos do domínio e sua integração à ontologia resultante, a maioria dos entrevistados informaram que tiveram dificuldades para obter esses conceitos. Alguns deles acham que a LPO inibiu a geração de novos conceitos na ontologia. Entretanto, também enfatizaram que a baixa experiência no desenvolvimento de ontologias e no domínio tratado pode ter influenciado na identificação dos conceitos específicos do domínio.

Com relação à notação visual de LPOs, os entrevistados destacaram que o modelo do processo foi um guia no desenvolvimento da ontologia de domínio, tornando esse processo claro e intuitivo. Porém, a maioria dos entrevistados teve dificuldades em relação ao símbolo adotado para representar padrões variantes. Além disso, dois grupos ressaltaram a necessidade de analisar a nomenclatura dos padrões. Eles explicitaram que nomes curtos facilitam a discussão sobre os padrões durante o desenvolvimento da ontologia, porém siglas não sugerem o significado do padrão e, portanto, deveriam ser evitadas. Por fim, alguns

entrevistados ressaltaram que tiveram dúvidas sobre a obrigatoriedade ou não de se seguir os fluxos nos caminhos indicados no processo.

4.4 Discussões

Conforme estabelecido no planejamento do estudo, seu objetivo foi avaliar o uso de uma LPO no desenvolvimento de ontologias de domínio e a notação visual adotada. Para isso, são considerados cinco indicadores: (i) Produtividade do desenvolvimento de ontologias; (ii) Qualidade da ontologia resultante; (iii) Grau de dificuldade para entendimento do mecanismo de funcionamento da LPO; (iv) Grau de dificuldade para entendimento dos símbolos da LPO; e (v) Ausência de Problemas de Representação da Notação.

Em relação ao uso de uma LPO no desenvolvimento de ontologias, todos os participantes afirmaram haver mais benefícios em se utilizar a LPO do que criar uma ontologia sem esse recurso e destacaram a diminuição no tempo de desenvolvimento da ontologia e melhoria da qualidade da ontologia resultante. Sendo assim, há indicação de melhoria para os indicadores Produtividade do desenvolvimento de ontologias e Qualidade da ontologia resultante, indicando que o uso de uma LPO impacta positivamente nesses aspectos do desenvolvimento de ontologias. Discussões detalhadas sobre as percepções obtidas a partir dos resultados dos questionários e entrevistas e, também, a partir da avaliação e correção das ontologias de domínio produzidas pelos participantes utilizando ISP-OPL podem ser encontradas em (RUY *et al.*, 2015a).

Com relação à notação visual adotada para representar a LPO, embora a maioria dos participantes tenha considerado fácil entender o mecanismo de funcionamento da LPO, o que significa um baixo grau de dificuldade para entendimento do mecanismo de funcionamento da LPO (indicador (c)), alguns participantes relataram ter tido dificuldades no entendimento de alguns símbolos, o que significa um maior grau de dificuldade para entendimento dos símbolos adotados (indicador (d)). Além disso, vários problemas foram relatados em relação aos símbolos utilizados e à necessidade de se representar melhor algumas situações, o que significa que há problemas na representação (indicador (e)). Esses resultados apontam para a necessidade de definição de símbolos mais expressivos e de tratamento de situações ambíguas (por exemplo, a obrigatoriedade ou não dos fluxos).

Assim, considerando-se os resultados, pode-se concluir que, no contexto da avaliação realizada nesse experimento, o uso de LPOs contribui para o aumento da produtividade do processo de desenvolvimento de ontologias de domínio e para a qualidade das ontologias resultantes, porém a notação utilizada para representar a LPO utilizada no estudo precisa de melhorias.

4.5 Ameaças à Validade

Ao se realizar um estudo é preciso levar em consideração as ameaças à sua validade. Elas devem ser tratadas na medida do possível e devem ser consideradas juntamente com os resultados obtidos no estudo. As ameaças relacionadas a este estudo foram divididas em categorias e são apresentadas a seguir.

Validade Interna: é definida como a capacidade de um novo estudo repetir o comportamento do estudo atual com os mesmos participantes e objetos com que ele foi realizado (BARROS *et al.*, 2002). A principal ameaça à validade interna é a comunicação e o compartilhamento de informações entre participantes do estudo. Para tratar essa ameaça, os participantes foram orientados a não trocar informações durante o preenchimento do questionário de avaliação.

Validade Externa: essa ameaça está relacionada à capacidade de repetir o mesmo comportamento com grupos diferentes daquele que participou do estudo (BARROS *et al.*, 2002). Nesse contexto, são consideradas ameaças (i) o fato dos participantes serem todos alunos do Programa de Pós-Graduação em Informática, o que poderia representar conhecimento limitado sobre ontologias e LPOs; e (ii) o fato de o estudo ter sido realizado em ambiente acadêmico. Para contornar a ameaça (i) foram identificados participantes com experiência em modelagem conceitual, tendo participado pelo menos de uma disciplina ou curso sobre o assunto. Além disso, participantes com perfis diferentes (alguns mais experientes, outros menos) foram considerados. Quanto à ameaça do item (ii), o fato de o estudo ser realizado em ambiente acadêmico faz com que ele apresente diferenças se comparado a um estudo semelhante realizado fora desse ambiente, como, por exemplo, o desenvolvimento das ontologias foi realizado em grupo e no contexto de um trabalho de uma disciplina. Ainda que alguns dos participantes tenham experiência e visão de projetos acadêmicos e não acadêmicos, não é possível eliminar a ameaça de validade externa relacionada a esse item.

Validade de Constructo: refere-se à relação entre os instrumentos e os participantes do estudo e a teoria que está sendo provada (BARROS *et al.*, 2002). Foram identificadas duas ameaças: (i) a ameaça de o participante dar respostas que não refletem a realidade devido a expectativas pessoais e por imaginar que ele próprio está sendo submetido à avaliação; e (ii) o participante manipular suas respostas de modo que elas sejam mais benéficas ao

pesquisador por serem colegas na mesma instituição de ensino. Para minimizar a possibilidade de ocorrer a ameaça (i), os participantes foram informados de que o experimento não representa qualquer tipo de avaliação pessoal, mas sim avaliação da LPO. Também foi assegurado o anonimato das respostas. Em relação à ameaça (ii), os participantes também foram informados de que deveriam realizar uma avaliação imparcial e deveriam realizar comentários críticos sobre as questões da avaliação de modo que eles pudessem ser utilizados para melhorias futuras da LPO.

Validade de Conclusão: mede a relação entre os tratamentos e os resultados e afere a capacidade do estudo em gerar conclusões (BARROS *et al.*, 2002). Foram identificadas as seguintes ameaças: (i) a pequena quantidade de participantes; e (ii) o fato de a maioria dos participantes ser iniciante em desenvolvimento de ontologias. Essas ameaças limitam a possibilidade de generalização dos resultados obtidos e do comportamento observado. Por isso, os resultados do estudo não podem ser generalizados e são considerados apenas resultados preliminares e indícios, não sendo conclusivos.

4.6 Considerações Finais

Neste capítulo foi apresentado um estudo experimental realizado com o objetivo de avaliar o uso de uma LPO no desenvolvimento de ontologias de domínio e a notação visual utilizada até então para representar LPOs.

Os resultados do estudo corroboram as afirmações encontradas na literatura de que LPOs auxiliam na engenharia de ontologias, melhorando a produtividade do processo e a qualidade das ontologias resultantes. Os resultados também apontaram para uma série de problemas na notação visual utilizada, dentre eles: dificuldades para identificar o nome dos padrões, falta de informação sobre a obrigatoriedade ou não de seguir caminhos determinados por fluxos no processo da LPO e dificuldade para entender o símbolo usado para representar padrões variantes.

Esses resultados podem ser entendidos como indícios de que a proposta de uma notação visual para LPOs é viável, visto que o uso de LPOs foi apontado como uma abordagem positiva no desenvolvimento de ontologias de domínio e que ainda há a necessidade de melhorias na sua representação.

Capítulo 5

Em Direção a uma Notação Visual para LPOs:

Primeiros Passos

Este capítulo apresenta os primeiros passos realizados para a definição da notação visual proposta neste trabalho. Na Seção 5.1 é apresentada uma versão preliminar à notação visual proposta. Na Seção 5.2 é apresentada S-OPL (Services Ontology Pattern Language), uma LPO desenvolvida utilizando-se a notação preliminar. Na Seção 5.3 é apresentado um estudo experimental que avaliou S-OPL a fim de identificar problemas e oportunidades de melhoria para a notação utilizada. Na Seção 5.4 são apresentadas as considerações finais do capítulo.

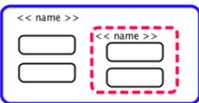
5.1 Notação Visual para LPOs: Uma Versão Preliminar

Visando-se adquirir conhecimento sobre o desenvolvimento de uma LPO e sua representação, decidiu-se por realizar uma experiência prática e desenvolver uma LPO a partir de uma ontologia de núcleo (*core ontology*) utilizando-se a notação usada até então nas LPOs desenvolvidas em trabalhos realizados no NEMO. Porém, embora algumas LPOs tivessem sido desenvolvidas no NEMO até então, havia diversas inconsistências na notação usada. Por exemplo, a representação visual para padrões variantes difere de uma LPO para outra e não fica claro qual símbolo deve ser usado. Dessa forma, antes de definir uma LPO usando a notação das LPOs desenvolvidas até então, foi preciso estabelecer quais elementos fariam parte da notação a ser usada (por exemplo, qual deve ser o símbolo para representar padrões variantes?). Assim, definiu-se que os símbolos a serem utilizados para representar o processo de aplicação de padrões da LPO seriam os apresentados na Tabela 5.1.

Tabela 5.1 - Elementos para representação visual do processo de LPOs.

Elemento	Representação gráfica	Semântica
Ponto de Entrada	●	Indica os pontos de entrada na LPO, isto é, indica os padrões na LPO que podem ser utilizados sem que seja necessário utilizar outros padrões antes.
Padrão	<code><<name>></code>	Representa os padrões ontológicos de domínio.
Fluxo de Controle	→	Representa um fluxo opcional. Apenas o elemento de fluxo com <code><<mandatory>></code> indica a obrigatoriedade de seguir o fluxo.
Nó de Decisão	◇	Os caminhos de saída indicam caminhos alternativos. Assim, é possível escolher apenas um dos caminhos de saída.

Tabela 5.1 - Elementos para representação visual do processo de LPOs (cont.).

Elemento	Representação gráfica	Semântica
Nó de Bifurcação (<i>Fork Node</i>)		Os caminhos de saída representam caminhos paralelos opcionais, ou seja, é possível seguir quaisquer caminhos de saída.
Nó de Junção (<i>Join Node</i>)		Os caminhos de entrada representam dependência múltipla, indicando que para seguir o caminho de saída, é necessário, antes, seguir todos os caminhos de entrada do nó.
Grupo de Padrões		Agrupa padrões ontológicos e grupo de padrões.
Grupo de Padrões Variantes		Agrupa os padrões variantes (padrões que resolvem o mesmo problema, mas de maneiras diferentes e mutuamente exclusivas). Desse conjunto de padrões, apenas um deles pode ser selecionado.

5.2 Service Ontology Pattern Language (S-OPL)

Uma vez estabelecida a notação visual a ser usada, foi realizado o desenvolvimento de uma LPO. O desenvolvimento da LPO permitiu avaliar a notação usada e também propiciou a aquisição de conhecimento sobre o desenvolvimento e representação de LPOs, o que foi útil para a evolução da notação preliminar na notação proposta neste trabalho, que será apresentada no Capítulo 6.

A LPO desenvolvida foi S-OPL (*Service Ontology Pattern Language*), que foi criada a partir de UFO-S (*Unified Foundational Ontology for Services*) (NARDI *et al.*, 2013), uma ontologia de núcleo de serviços baseada em compromissos e fundamentada na *Unified Foundational Ontology* (UFO) (GUIZZARDI, 2005).

A conceituação de UFO-S é baseada no estabelecimento e cumprimento de compromissos e reivindicações entre os participantes do serviço (i.e., prestadores de serviços e clientes) ao longo do ciclo de vida do serviço. UFO-S concentra-se nas três fases principais do ciclo de vida de serviços, a saber: Oferta de Serviços, Negociação de Serviços e Entrega de Serviços.

Para o desenvolvimento da LPO, a *core ontology* foi fragmentada de modo a solucionar problemas abordados pelo seu domínio. Esses fragmentos formaram os padrões ontológicos da LPO: *Service Ontology Pattern Language* (S-OPL) (FALBO *et al.*, 2016 e (QUIRINO *et al.*,

2015). Os padrões extraídos de UFO-S foram organizados em S-OPL em quatro grupos: Oferta de Serviço, Negociação e Acordo de Serviço, Entrega de Serviço, Provedor e Cliente de Serviço. Os padrões de S-OPL e seu processo são apresentados nas próximas seções.

5.2.1 Padrões Ontológicos de S-OPL

A seguir os aspectos tratados em cada grupo de S-OPL e os padrões neles contidos são brevemente apresentados.

Oferta de Serviço (*Service Offering*)

Uma relação de serviço inicia-se com uma oferta de serviço, que é estabelecida entre um prestador de serviços e uma comunidade alvo, cujos membros são ditos clientes alvo. A oferta de serviços compreende um conjunto de compromissos de um prestador de serviços para uma comunidade alvo e as correspondentes reivindicações da comunidade alvo para o prestador de serviços. A Tabela 5.2 lista os padrões deste grupo.

Tabela 5.2 - Padrões do Grupo Oferta do Serviço.

Id	Nome	Descrição
SOffering	Oferta de Serviço (<i>Service Offering</i>)	Representa as ofertas de serviço estabelecidas pelos prestadores de serviços para as comunidades alvo, e respectivamente, para os membros dessas comunidades.
SODescription	Descrição da Oferta do Serviço (<i>Service Offering Description</i>)	Permite detalhar ofertas de serviços por meio de descrições de oferta de serviços.
SOCcommitments	Compromissos da Oferta do Serviço (<i>Service Offering Commitments</i>)	Representa os compromissos do prestador de serviços para com a comunidade alvo no contexto da oferta do serviço.
SOClaims	Reivindicações da Oferta do Serviço (<i>Service Offering Claims</i>)	Representa as reivindicações da comunidade alvo para com o prestador de serviços no contexto da oferta do serviço.

Negociação e Acordo de Serviço (*Service Negotiation and Agreement*)

Uma vez que um serviço é ofertado, pode ocorrer a negociação de serviços que, em geral, é motivada pelo interesse de um cliente alvo na oferta de serviço. Durante a negociação de serviço, prestadores de serviços e clientes alvo interagem a fim de estabelecer um acordo resguardando-se dos compromissos e reivindicações envolvidos. Uma negociação de serviço bem-sucedida resulta em um acordo de serviço no qual o prestador do serviço desempenha o papel de prestador de serviços contratado e o cliente-alvo o de cliente do serviço. Um acordo de serviço é composto de compromissos e reivindicações estabelecidos entre o prestador e o cliente. Informações sobre os compromissos e reivindicações, bem como sobre

o serviço contratado podem ser registradas na descrição do acordo de serviço (por exemplo, um contrato). A Tabela 5.3 apresenta os padrões deste grupo.

Tabela 5.3 - Padrões do Grupo Negociação e Acordo de Serviços.

Id	Nome	Descrição
SNegotiation	Negociação de Serviço (Service Negotiation)	Representa uma Negociação de Serviço e a Oferta de Serviço na qual a negociação se resguarda, sem abordar um possível acordo resultante dele.
SNegAgree	Negociação e Acordo de Serviço (Service Negotiation and Agreement)	Representa uma Negociação de Serviço e o Acordo de Serviço que possivelmente resulta dele, considerando também a Oferta de Serviço correspondente.
SOfferAgree	Oferta e Acordo do Serviço (Service Offering and Agreement)	Representa um Acordo de Serviço em conformidade com uma Oferta de Serviço, sem abordar aspectos de negociação de serviço.
SAgreement	Acordo de Serviço (Service Agreement)	Representa um Acordo do Serviço, sem abordar aspectos da oferta e negociação do serviço.
SADescription	Descrição de Acordo de Serviço (Service Agreement Description)	Permite descrever Acordos de Serviços por meio de Descrições de Acordo de Serviço.
HPCommitments	Compromissos do Provedor de Serviços Contratado (Hired Provider Commitments)	Representa os compromissos de um Provedor de Serviços Contratado para com um Cliente de Serviço.
HPClaims	Reivindicações do Provedor de Serviços Contratado (Hired Provider Claims)	Representa as reivindicações de um Provedor de Serviços Contratado para com um Cliente de Serviço.
SCCommitments	Compromissos do Cliente de Serviço (Service Customer Commitments)	Representa os compromissos de um Cliente de Serviço para com um Provedor de Serviços Contratado.
SCClaims	Reivindicações do Cliente de Serviço (Service Customer Claims)	Representa as reivindicações de um Cliente de Serviço para com um Provedor de Serviços Contratado.

Entrega de Serviço (*Service Delivery*)

A entrega de serviços diz respeito à execução de ações destinadas a cumprir os compromissos estabelecidos no acordo de serviço. A entrega de serviços envolve ações realizadas pelo prestador de serviços contratado, pelo cliente e por ambos. Essas ações são motivadas pelos compromissos estabelecidos no acordo de serviço. A Tabela 5.4 apresenta os padrões deste grupo.

Tabela 5.4 - Padrões do grupo Entrega de Serviço.

Id	Nome	Objetivos
SDelivery	Entrega de Serviço (Service Delivery)	Representa as ações realizadas para cumprir os compromissos estabelecidos no Acordo de Serviço.
HPActions	Ações do Prestador de Serviços Contratado (Hired Service Provider Actions)	Representa as ações da Entrega do Serviço que são realizadas apenas pelo Prestador de Serviço Contratado.
SCActions	Ações do Cliente do Serviço (Service Customer Actions)	Representa as ações da Entrega do Serviço que são realizadas apenas pelo Cliente do Serviço.
Interations	Interações (Interations)	Representa as ações da Entrega do Serviço nas quais o Prestador de Serviço Contratado e o Cliente do Serviço agem em conjunto.
HPActionMotivation	Motivação da Ação do Prestador Contratado (Hired Provider Action Motivation)	Representa os relacionamentos entre as ações realizadas pelo Prestador de Serviço Contratado e os compromissos que as motivaram.
SCActionMotivation	Motivação da Ação do Cliente do Serviço (Service Customer Action Motivation)	Representa os relacionamentos entre as ações realizadas pelo Cliente do Serviço e os compromissos que as motivaram.
InteractionMotivation	Motivações e Interações (Motivations for Interactions)	Representa os relacionamentos entre as interações do Prestador de Serviço Contratado e o Cliente do Serviço e os compromissos que as motivaram.

Prestador de Serviço e Cliente (*Service Provider and Customer*)

Prestador de serviços é o papel desempenhado por agentes quando se comprometem com uma comunidade alvo em uma oferta de serviço. Cliente alvo é o papel desempenhado por agentes quando, como consequência de uma oferta de serviço, tornam-se membros de uma comunidade alvo. Quando um acordo de serviço é estabelecido, o prestador de serviços passa a desempenhar o papel de prestador de serviços contratado, enquanto o cliente alvo desempenha o papel de cliente.

Dependendo do serviço específico que está sendo modelado, esses papéis podem ser desempenhados por diferentes tipos de agentes, ou seja, pessoas, organizações e unidades organizacionais. Os padrões deste grupo permitem que o engenheiro de ontologias escolha quais tipos de agentes efetivamente desempenharão esses papéis no domínio que está sendo modelado. A Tabela 5.5 apresenta os padrões.

Tabela 5.5 – Padrões do grupo Prestador de Serviço e Cliente.

Id	Name	Intent
P-Provider	Prestador de Serviço Pessoa (Person Provider)	Representa pessoas como prestadores de serviços.
O-Provider	Prestador de Serviço Organização (Organization Provider)	Representa organizações como prestadores de serviço.
OU-Provider	Prestador de Serviço Unidade Organizacional (Organizational Unit Provider)	Representa unidades organizacionais como prestadores de serviço.
O-OU-Provider	Prestador de Serviço Organização / Unidade Organizacional (Organization / Organizational Unit Provider)	Representa organizações e unidades organizacionais como prestadores de serviço.
P-O-Provider	Prestador de Serviço Pessoa / Organização (Person / Organization Provider)	Representa pessoas e organizações como prestadores de serviço.
P-OU-Provider	Prestador de Serviço Pessoa / Unidade Organizacional (Person / Organizational Unit Provider)	Representa pessoas e unidades organizacionais como prestadores de serviço.
P-O-OU-Provider	Prestador de Serviço Pessoa / Organização / Unidade Organizacional (Person / Organization / Organizational Unit Provider)	Representa pessoas, organizações e unidades organizacionais como prestadores de serviço.
P-TCustomer	Cliente-Alvo Pessoa (Person Target Customer)	Representa pessoas como clientes-alvo.
O-TCustomer	Cliente-Alvo Organização (Organization Target Customer)	Representa organizações como clientes-alvo.
OU-TCustomer	Cliente-Alvo Unidade Organizacional (Organizational Unit Target Customer)	Representa unidades organizacionais como clientes-alvo.
O-OU-TCustomer	Cliente-Alvo Organização / Unidade Organizacional (Organization / Organizational Unit Target Customer)	Representa organizações e unidades organizacionais como clientes-alvo.
P-O-TCustomer	Cliente-Alvo Pessoa / Organização (Person / Organization Target Customer)	Representa pessoas e organizações como clientes-alvo.
P-OU-TCustomer	Cliente-Alvo Pessoa / Unidade Organizacional (Person / Organizational Unit Target Customer)	Representa pessoas e unidades organizacionais como clientes-alvo.
P-O-OU-TCustomer	Cliente-Alvo Pessoa / Organização / Unidade Organizacional (Person / Organization / Organizational Units Target Customer)	Representa pessoas, organizações e unidades organizacionais como clientes-alvo.
P-OU-Provider	Prestador de Serviço Pessoa / Unidade Organizacional (Person / Organizational Unit Provider)	Representa pessoas e unidades organizacionais como prestadores de serviço.

Tabela 5.5 – Padrões do grupo Prestador de Serviço e Cliente (cont.).

Id	Name	Intent
P-OU-Provider	Prestador de Serviço Pessoa / Unidade Organizacional (Person/ Organizational Unit Provider)	Representa pessoas e unidades organizacionais como prestadores de serviço.
P-O-OU-Provider	Prestador de Serviço Pessoa / Organização / Unidade Organizacional (Person/ Organization / Organizational Unit Provider)	Representa pessoas, organizações e unidades organizacionais como prestadores de serviço.
P-TCustomer	Cliente-Alvo Pessoa (Person Target Customer)	Representa pessoas como clientes-alvo.
O-TCustomer	Cliente-Alvo Organização (Organization Target Customer)	Representa organizações como clientes-alvo.
OU-TCustomer	Cliente-Alvo Unidade Organizacional (Organizational Unit Target Customer)	Representa unidades organizacionais como clientes-alvo.
O-OU-TCustomer	Cliente-Alvo Organização / Unidade Organizacional (Organization / Organizational Unit Target Customer)	Representa organizações e unidades organizacionais como clientes-alvo.
P-O-TCustomer	Cliente-Alvo Pessoa / Organização (Person / Organization Target Customer)	Representa pessoas e organizações como clientes-alvo.
P-OU-TCustomer	Cliente-Alvo Pessoa / Unidade Organizacional (Person / Organizational Unit Target Customer)	Representa pessoas e unidades organizacionais como clientes-alvo.
P-O-OU-TCustomer	Cliente-Alvo Pessoa / Organização / Unidade Organizacional (Person / Organization / Organizational Units Target Customer)	Representa pessoas, organizações e unidades organizacionais como clientes-alvo.
P-HProvider	Prestador de Serviço Contratado Pessoa (Person Hired Provider)	Representa pessoas como prestadores de serviço contratado.
O-HProvider	Prestador de Serviço Contratado Organização (Organization Hired Provider)	Representa organizações como prestadores de serviço contratado.
OU-HProvider	Prestador de Serviço Contratado Unidade Organizacional (Organizational Unit Hired Provider)	Representa unidades como prestadores de serviço contratado.
O-OU-HProvider	Prestador de Serviço Contratado Organização / Unidade Organizacional (Organization / Organizational Unit Hired Provider)	Representa organizações e unidades organizacionais como prestadores de serviço contratado.
P-O-HProvider	Prestador de Serviço Contratado Pessoa / Organização (Person / Organization Hired Provider)	Representa pessoas e organizações como prestadores de serviço contratado.
P-OU-HProvider	Prestador de Serviço Contratado Pessoa / Unidade Organizacional (Person/ Organizational Unit Hired Provider)	Representa pessoas e unidades organizacionais como prestadores de serviço contratado.
P-O-OU-HProvider	Prestador de Serviço Contratado Pessoa / Organização / Unidade Organizacional (Person/ Organization / Organizational Unit Hired Provider)	Representa pessoas, organizações e unidades organizacionais como prestadores de serviço contratado.

Tabela 5.5 – Padrões do grupo Prestador de Serviço e Cliente (cont.).

Id	Name	Intent
P-Customer	Cliente Pessoa (<i>Person Customer</i>)	Representa pessoas como clientes de serviço.
O-Customer	Cliente Organização (<i>Organization Customer</i>)	Representa organizações como clientes de serviço.
OU-Customer	Cliente Unidade Organizacional (<i>Organizational Unit Customer</i>)	Representa unidades organizacionais como clientes de serviço.
O-OU-Customer	Cliente Organização / Unidade Organizacional (<i>Organization / Organizational Unit Customer</i>)	Representa organizações e unidades organizacionais como clientes de serviço.
P-O-Customer	Cliente Pessoa / Organização (<i>Person / Organization Customer</i>)	Representa pessoas e organizações como clientes de serviço.
P-OU-Customer	Cliente Pessoa / Unidade Organizacional (<i>Person / Organizational Unit Customer</i>)	Representa pessoas e unidades organizacionais como clientes de serviço.
P-O-OU-Customer	Cliente Pessoa / Organização / Unidade Organizacional (<i>Person / Organization / Organizational Units Customer</i>)	Representa pessoas, organizações e unidades organizacionais como clientes de serviço.

5.2.2 Modelo de Processo de S-OPL

A Figura 5.1 apresenta o modelo de processo de S-OPL (FALBO *et al.*, 2016). Na notação utilizada, conforme descrito na primeira seção deste capítulo, padrões são representados por nós de ação (retângulos arredondados rotulados). Grupos de padrões são delimitados por retângulos azuis com bordas arredondadas. Nós iniciais (*entry points*), representados por círculos sólidos, são usados para indicar os pontos de entrada na LPO, isto é, padrões na linguagem que podem ser utilizados sem que seja necessário utilizar outros padrões antes. Nós de bifurcação (*fork nodes*) (segmentos de linha com vários fluxos de saída) são usados para representar caminhos paralelos opcionais (se o engenheiro de ontologias decide seguir o caminho de entrada do nó de bifurcação, então ele pode seguir quaisquer caminhos de saída). Nós de junção (*join nodes*) (segmentos de linha com múltiplos fluxos de entrada) são usados para representar dependência múltipla, indicando que para seguir o caminho de saída do nó de junção, o engenheiro de ontologias deve, antes, seguir todos os caminhos de entrada do nó. Nós de decisão (*decision nodes*), representados por um losango, são usados para representar caminhos alternativos. Assim, se o engenheiro de ontologias decide seguir o caminho de entrada do nó de decisão, ele deve escolher um e apenas um dos caminhos de saída do nó de decisão. Grupos de padrões variantes (*variant patterns*), representados por retângulos arredondados com linhas pontilhadas vermelhas, são um conjunto de padrões que resolvem o mesmo problema, mas de maneiras diferentes e

mutuamente exclusivas. Assim, desse conjunto de padrões, apenas um deles pode ser selecionado. Finalmente, fluxos de controle (linhas com setas) representam as sequências admissíveis de caminhos que o engenheiro de ontologias pode seguir na LPO. Fluxos de controle são opcionais, ou seja, o engenheiro de ontologias pode decidir seguir ou não os fluxos, dependendo do escopo da ontologia a ser desenvolvida. Assim, o engenheiro de ontologias pode selecionar um determinado padrão e decidir não usar qualquer outro depois disso, mesmo se houver fluxos de controle desse padrão para outros. No entanto, quando um fluxo de controle é estereotipado com `<<mandatory>>`, o caminho deve ser obrigatoriamente seguido. Na Figura 5.1, são usadas cores diferentes para identificar padrões de diferentes grupos.

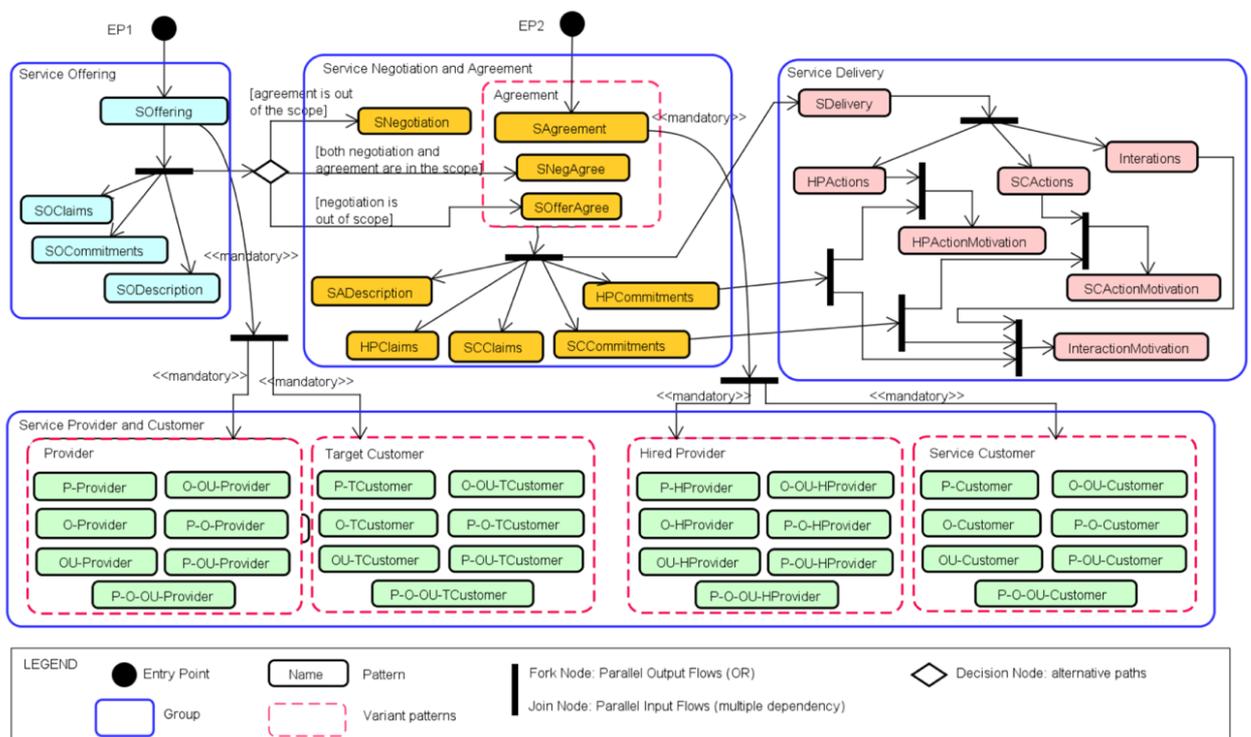


Figura 5.1 - Modelo do Processo de S-OPL (versão inicial).

Como mostra a Figura 5.1, S-OPL tem dois pontos de entrada: EP1 e EP2. O engenheiro de ontologias deve escolher um deles, dependendo do escopo da ontologia de serviço específica a ser desenvolvida. Quando os requisitos para a ontologia sendo desenvolvida incluem a descrição da oferta de serviços, então o ponto de partida é EP1. Caso contrário, o ponto de partida é EP2.

Quando EP1 é escolhido, o engenheiro de ontologias deve usar primeiro o padrão *SOffering* para modelar a oferta do serviço. Em seguida, o engenheiro de ontologias deve seguir o primeiro dos caminhos obrigatórios, o que leva ao grupo Prestador de Serviço e

Cliente (*Service Provider and Customer*), para a modelagem dos tipos de prestadores de serviço e clientes-alvo envolvidos na oferta. Prestadores de serviço e clientes-alvo podem ser pessoas, organizações ou unidades organizacionais. Portanto, o engenheiro de ontologias deve selecionar um dos padrões do grupo Prestador de Serviço (*Provider*), e um dos padrões do grupo Clientes-Alvo (*Target Customer*). Uma vez modelados prestadores de serviço e clientes-alvo, o engenheiro de ontologias pode seguir o caminho que leva ao primeiro nó de bifurcação do processo. A partir desse nó há vários caminhos de saída, que são opcionais. O engenheiro de ontologias pode utilizar os seguintes padrões: *SOClaims* e *SOCCommitments*, se o engenheiro de ontologias está interessado em modelar, respectivamente, as reivindicações e os compromissos envolvidos na oferta do serviço; e *SODescription*, se o engenheiro de ontologias está interessado em tratar descrição de oferta de serviço.

Uma vez modelada a oferta de serviços, o engenheiro de ontologias pode resolver problemas relacionados a negociação e acordo de serviço. Caso a oferta de serviços esteja fora do escopo da ontologia, EP2 é o ponto de entrada, permitindo modelar acordo sem tratar problemas relacionados a oferta.

Se o engenheiro de ontologias modelou previamente a oferta de serviço, ele deve decidir primeiro se precisa representar a negociação e o acordo do serviço. Se estiver interessado apenas em representar a negociação do serviço (o acordo está fora do escopo da ontologia), ele deve usar o padrão *SNegotiation*, um padrão que captura apenas a negociação dos serviços e sua relação com a oferta de serviços ao qual a negociação se refere. Se ele está interessado em representar tanto a negociação do serviço quanto o acordo resultante, ele deve usar *SNegAgree*, um padrão que modela a negociação de serviços, o acordo de serviço e suas relações com a oferta de serviço correspondente. Finalmente, se ele está interessado em modelar apenas o acordo de serviço e sua conformidade com uma oferta de serviço (negociação está fora do escopo da ontologia), ele deve escolher o padrão *SOfferAgree*, o que representa um acordo em conformidade com uma oferta.

Se EP2 é o ponto de entrada (oferta de serviço, e, assim, negociação de serviços, estão fora do escopo da ontologia), o primeiro padrão a ser utilizado é *SAgreement*. Em seguida, o engenheiro de ontologias deve selecionar um dos padrões do grupo Prestador de Serviço Contratado (*Hired Provider*), e um dos padrões do grupo Cliente (*Service Customer*) para modelar os possíveis tipos de prestador de serviço contratado e cliente do serviço. Note que isto é necessário somente se o ponto de entrada é EP2, uma vez que, quando ponto de entrada é EP1, os tipos de fornecedores e clientes-alvo já foram modelados.

Uma vez que o acordo do serviço tenha sido modelado, os seguintes padrões podem ser usado opcionalmente: *HPCommitments* e *HPClaims*, se o engenheiro de ontologias está interessado em modelar os compromissos e reivindicações do prestador de serviço contratado, respectivamente; *SCCommitments* e *SCClaims*, se ele está interessado em modelar os compromissos e reivindicações dos clientes de serviço, respectivamente; e *SADescription*, se há interesse em descrever o contrato do serviço por meio de uma descrição.

Após a modelagem do acordo do serviço, o engenheiro de ontologias pode modelar a entrega de serviços. O primeiro padrão a ser utilizado é *SDelivery*. Em seguida, pode-se modelar as ações envolvidas em uma entrega. Para tanto, os seguintes padrões devem ser aplicados: *HPActions*, para modelar as ações realizadas pelo prestador de serviço contratado; *SCActions*, para modelar as ações executadas pelo cliente; e *Interactions*, para modelar as ações realizadas por ambos, em conjunto. Uma vez modelada as ações, é possível modelar as relações entre as ações e os compromissos que as motivaram, usando os seguintes padrões: *HPActionMotivation*, *SCActionMotivation* e *InteracitonMotivation*.

Após o desenvolvimento de S-OPL, ela foi aplicada em um estudo de caso real no domínio de Tecnologias de Informação e Comunicação em uma empresa italiana. Os resultados desse estudo foram registrados em (QUIRINO *et al.*, 2015) e (FALBO *et al.*, 2016).

5.3 Estudo Experimental para Avaliação de S-OPL e da Notação Visual Adotada

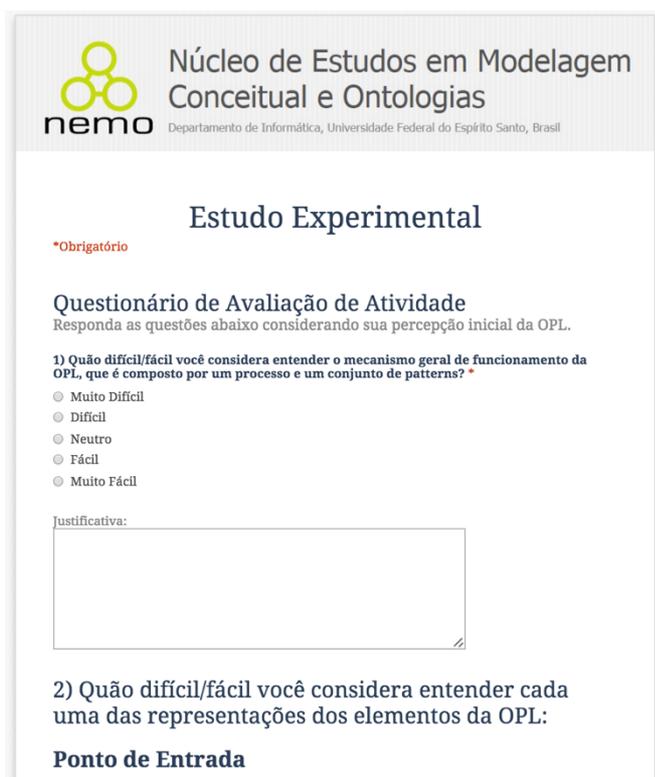
Para avaliar a notação visual adotada, foi conduzido um estudo experimental. O estudo foi realizado com alunos do Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo e ocorreu no contexto da disciplina Engenharia de Ontologias ministrada no segundo semestre de 2015.

O estudo teve o mesmo **objetivo** do primeiro experimento conduzido no contexto deste trabalho (vide Capítulo 4), ou seja, avaliar o uso de uma LPO no desenvolvimento de ontologias de domínio e a notação visual adotada. S-OPL foi a LPO utilizada no estudo.

O estudo experimental seguiu o mesmo **procedimento** usado no primeiro experimento (apresentação de S-OPL, desenvolvimento de ontologias de domínio e aplicação de questionário e entrevista) e utilizou a mesma **instrumentação** (termo de consentimento, formulário de caracterização de perfil e questionário). Utilizando S-OPL, os participantes produziram ontologias para os domínios dos serviços de Administração de Condomínios, TV por Assinatura, Limpeza, Hospedagem em Hotéis e Transportes de

Passageiros (Taxi/Uber). Após a coleta e análise dos dados dos questionários e entrevistas, a notação visual foi alterada obtendo-se a notação visual que será descrita no Capítulo 6.

O **questionário** usado nesse estudo experimental foi o mesmo usado no primeiro experimento e inclui questões relacionadas ao uso da LPO no desenvolvimento de ontologias e à representação visual adotada. Para elaboração do questionário eletrônico, neste experimento foi utilizado o *Google Forms*¹. Os formulários utilizados podem ser vistos na íntegra no Apêndice A. A Figura 5.2 apresenta um fragmento do questionário utilizado nesse estudo.



Logo do Núcleo de Estudos em Modelagem Conceitual e Ontologias (nemo) com o endereço: Departamento de Informática, Universidade Federal do Espírito Santo, Brasil.

Estudo Experimental

***Obrigatório**

Questionário de Avaliação de Atividade

Responda as questões abaixo considerando sua percepção inicial da OPL.

1) **Quão difícil/fácil você considera entender o mecanismo geral de funcionamento da OPL, que é composto por um processo e um conjunto de patterns? ***

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Justificativa:

2) **Quão difícil/fácil você considera entender cada uma das representações dos elementos da OPL:**

Ponto de Entrada

Figura 5.2 - Fragmento do Questionário de Avaliação usado.

Os **participantes** desse estudo foram 9 alunos do Programa de Pós-Graduação em Informática, da Universidade Federal do Espírito Santo, com conhecimentos básicos de modelagem conceitual. Dentre os participantes, 6 declararam ter experiência *média* com modelagem conceitual. Os demais declararam ter *baixa* experiência. Em relação à experiência com desenvolvimento de ontologias, 2 participantes declararam ter *média* experiência, 2 declararam ter experiência *baixa* e 5 declararam ter *nenhuma* experiência. Por fim, 4 participantes responderam ter *baixa* experiência com LPOs, tendo sido ela adquirida durante

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

o primeiro estudo experimental (vide Capítulo 4), que foi realizado no contexto de uma disciplina cursada anteriormente e 5 declararam não possuir experiência com LPOs.

Embora a maioria dos participantes do estudo não tenha experiência com LPOs ou com desenvolvimento de ontologias, eles não foram excluídos da pesquisa, uma vez que todos possuem um conhecimento mínimo em modelagem conceitual e refletem os possíveis perfis de usuários de LPOs. De toda forma, a avaliação dos resultados deve levar em consideração o perfil dos participantes do estudo.

5.3.1 Resultados

Nesta seção são apresentados os principais resultados obtidos a partir dos questionários respondidos pelos participantes.

5.3.1.1 Percepção acerca da LPO

O questionário inicia-se com perguntas relacionadas à LPO propriamente dita, visando capturar a percepção dos participantes quanto ao entendimento do funcionamento da LPO e dos símbolos utilizados na sua notação visual.

Embora os participantes do estudo tenham *baixa* ou *nenhuma* experiência anterior com LPOs, nenhum deles considerou o mecanismo geral de funcionamento da LPO, que é composto por um processo e um conjunto de padrões, *muito difícil*. A maioria dos participantes (8) considerou *fácil* e apenas 1 considerou *difícil*. Esse participante declarou ter *baixa* experiência com modelagem conceitual e *nenhuma* experiência com ontologias e LPOs, sendo a aplicação S-OPL nesse estudo a primeira experiência dele com ontologias e LPOs. Dentre os alunos que consideraram o mecanismo de funcionamento da LPO *fácil*, 2 declararam ter *baixa* experiência com modelagem conceitual e 6 declararam ter experiência *média*. Em relação a ontologias, 4 participantes informaram ter *nenhuma* experiência, tendo apenas 2 com experiência média e 2 com experiência baixa. Quanto à experiência com LPOs, 4 responderam ter *baixa* experiência e os outros 4 declararam ter *nenhuma* experiência. Esses resultados são indícios de que não é necessário ter muita experiência com ontologias ou LPOs para entender o mecanismo de LPOs. A Figura 5.3 ilustra esses resultados.

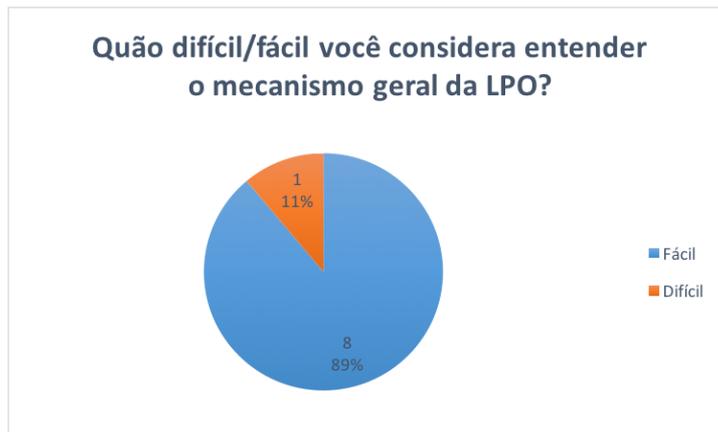


Figura 5.3 – Percepção dos participantes quanto ao entendimento do mecanismo da LPO.

Com relação aos símbolos utilizados para representar os elementos da LPO, houve uma boa aceitação por parte da maioria dos participantes. Um aluno considerou os símbolos usados para representar Grupo de Padrões e Grupo de Padrões Variantes de *difícil* compreensão. Esse participante declarou ter *baixa* experiência em modelagem conceitual e *nenhuma* experiência com ontologias e LPOs. Tanto ele quanto os que classificaram o grau de dificuldade de entendimento como *neutro* não justificaram suas respostas. A Figura 5.4 apresenta as respostas sobre esse tópico.

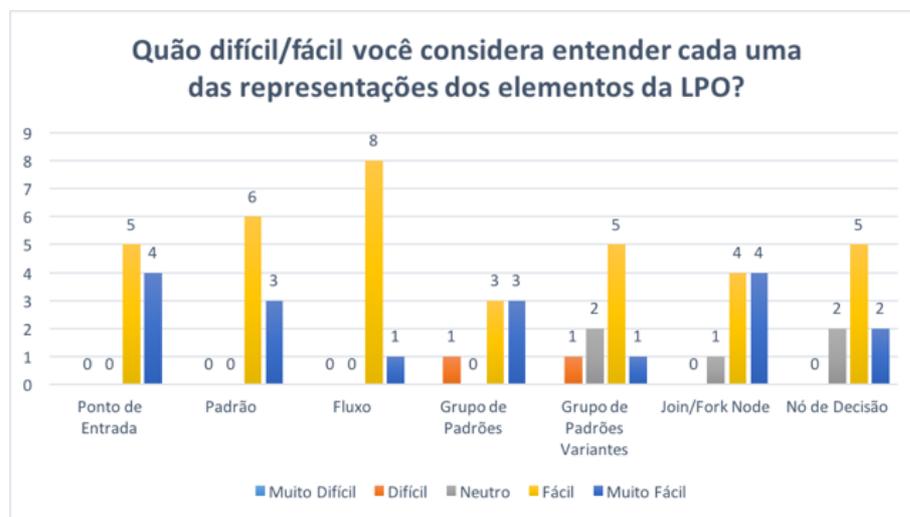


Figura 5.4 – Compreensão dos símbolos dos elementos de S-OPL.

O símbolo da notação visual que gerou maior dificuldade de entendimento foi o usado para representar Padrões Variantes. A justificativa apontada foi a dificuldade para entender a exclusividade mútua entre os padrões. Outro símbolo citado foi o *Fork Node*. Os participantes alegaram dificuldades em estabelecer a ordem a ser seguida na seleção das saídas, ou seja, definir a ordem dos caminhos alternativos. Além disso, sugeriram que as setas tivessem um estilo diferente, pois a linha sólida sugeriu a eles “obrigatoriedade” de seguir

todos os caminhos. Dúvidas surgiram em relação a escolher uma saída do *fork node* e depois ter que retornar as outras saídas do *fork node*. Para eles, esse retorno não é intuitivo. Por fim, dois participantes tiveram dificuldades sobre obrigatoriedade ou não de seguir os fluxos. A Figura 5.5 apresenta os símbolos que os participantes tiveram alguma dificuldade para entender.

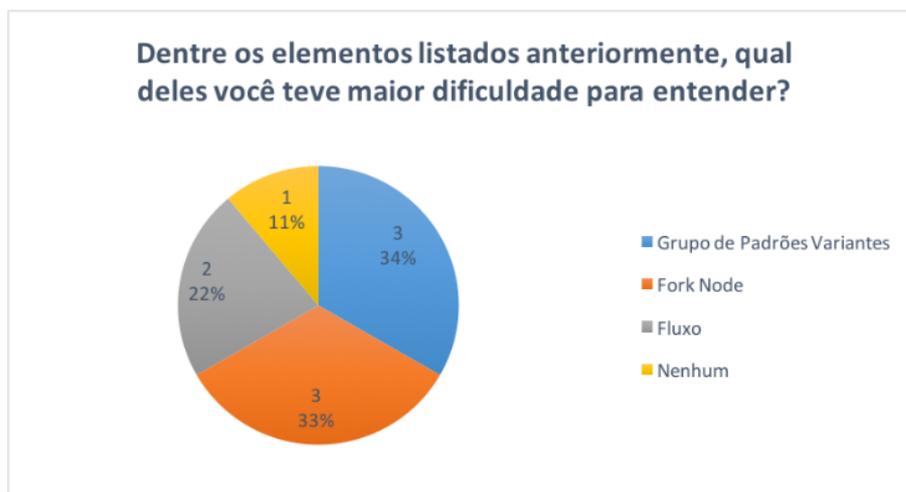


Figura 5.5 - Elementos mais difíceis de entender de S-OPL.

As sugestões de melhoria apresentadas pelos participantes da primeira fase focaram no elemento *Fork Node*, destacando-se: (i) adicionar informações nas saídas desse elemento, de modo a guiar o usuário na escolha do caminho a ser seguido; (ii) trocar o símbolo por algum mais intuitivo; e (iii) colocar as saídas diretamente nos padrões, possibilitando assim, a eliminação do símbolo.

5.3.1.2 Percepção acerca do Uso de LPOs no Desenvolvimento de Ontologias de Domínio

A segunda parte do questionário concentra-se no uso de uma LPO no desenvolvimento de ontologias.

No que se refere ao grau de dificuldade encontrado no desenvolvimento da ontologia de domínio com o uso da LPO, a maioria dos participantes considerou o desenvolvimento *fácil* (4 participantes) ou *muito fácil* (3 participantes). Dos participantes, um informou *neutro* e um considerou o processo *difícil*. Ambos justificaram suas respostas no fato de terem pouca experiência com ontologias e LPOs. Alguns participantes destacaram que o uso da LPO facilitou o desenvolvimento da ontologia de domínio por ser intuitivo, simples e por auxiliar na condução do raciocínio. A Figura 5.6 apresenta esses resultados.

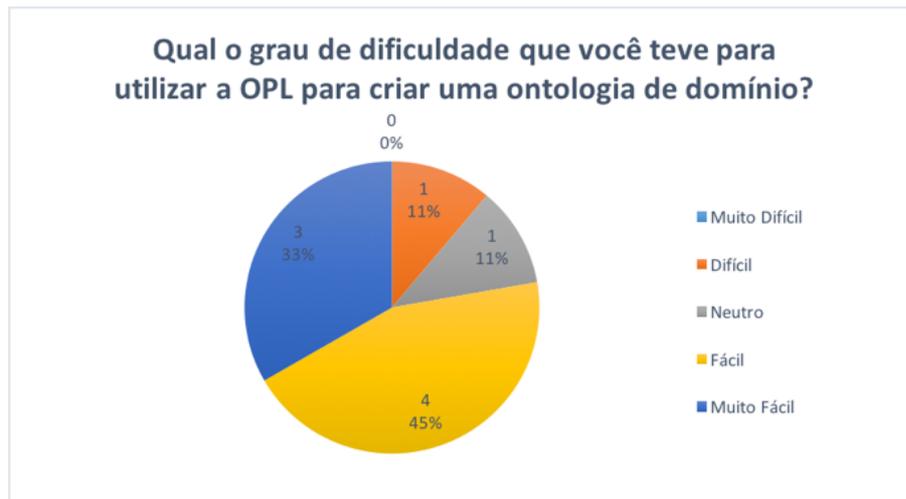


Figura 5.6 - Grau de dificuldade na utilização de LPOs.

Os quatro participantes com alguma experiência em desenvolvimento de ontologias responderam questões relativas a diferenças entre o desenvolvimento de ontologias a partir do zero e usando LPOs. Todos eles avaliaram positivamente o uso da LPO. Alguns consideraram que o agrupamento dos padrões por aspectos do domínio auxilia a manter o foco em problemas centrais sem perder a atenção em problemas adjacentes, o que ajudou na definição do escopo e minimizou o tempo que normalmente seria despendido no desenvolvimento de uma ontologia.

Em relação à qualidade da ontologia resultante, os quatro participantes afirmaram que o uso da LPO influenciou na qualidade da ontologia resultante. Alguns declararam que o desenvolvimento da ontologia com a LPO agilizou o processo e ajudou na organização das ideias. Um dos participantes justificou que a fundamentação em uma ontologia de núcleo influencia diretamente na qualidade da ontologia e que o uso da LPO facilitou o desenvolvimento e aprimorou a ontologia. Em resumo, os quatro participantes consideraram que o uso de LPOs contribuiu para a qualidade da ontologia resultante. Alguns justificaram que a LPO ajudou a identificar alguns conceitos que deveriam aparecer na ontologia, de modo que ela ficou completa e bem fundamentada.

Em relação às dificuldades encontradas no uso de LPOs, poucos participantes declararam ter enfrentado problemas na integração dos padrões durante o desenvolvimento da ontologia. Um dos participantes considerou difícil o passo de integrar um padrão em outro padrão. Outro participante ressaltou dificuldades no entendimento dos conceitos de domínio, que influenciou no momento de integrar os padrões.

Como sugestões de melhoria, os participantes sugeriram realizar alterações no *Foré Node* para deixá-lo mais intuitivo e aprimorar a documentação disponibilizada para cada

padrão, principalmente incluir definições mais detalhadas para os conceitos presentes nos padrões, bem como exemplos.

5.3.2 Entrevistas

Após a aplicação dos questionários foram realizadas entrevistas com os grupos de participantes do experimento com a finalidade de obter informações adicionais sobre sua percepção em relação à notação visual de LPOs e ao uso de LPOs no desenvolvimento de ontologias de domínio. As perguntas realizadas nas entrevistas são apresentadas no Apêndice A desta dissertação.

Em relação à notação visual de LPOs, alguns entrevistados destacaram que os símbolos utilizados eram claros, intuitivos e de fácil entendimento, principalmente por serem baseados em modelos da UML, notação da qual tinham algum conhecimento prévio. Alguns dos entrevistados tiveram dificuldades inicialmente em entender o símbolo do Grupo de Padrões Variantes, porém ao lerem a documentação, as dúvidas foram sanadas. Outro elemento destacado nas entrevistas foi o *Fork Node*. Muitos dos entrevistados consideraram difícil identificar as possíveis sequências dadas por esse elemento e não acharam intuitivo as saídas serem opcionais. O fato de a notação permitir seguir diversos fluxos gerou dúvidas sobre a sequência na qual utilizar os padrões, desencadeando também, perda do fluxo principal de execução do processo. Alguns entrevistados enfatizaram que ter retorno de fluxo nos elementos de um grupo para outro tornou o processo menos natural. Grande parte dos entrevistados ressaltaram dificuldades em entender a finalização da aplicação do processo. Ainda que a documentação informasse que a aplicação dos padrões poderia ser finalizada em qualquer ponto, os entrevistados afirmaram notar a ausência de um símbolo para indicar o término do processo de aplicação de padrões. Por fim, um dos grupos entrevistados destacou a importância das cores utilizadas nos padrões do modelo de processo. Eles afirmaram que esse recurso ajudou muito na percepção das divisões do domínio e facilitou a rastreabilidade de informações na documentação.

No que se refere ao processo de desenvolvimento da ontologia de domínio utilizando LPOs, os entrevistados consideraram ser um método mais eficiente do que o tradicional. Destacaram que as LPOs guiaram de forma prática e eficaz o usuário durante todo o processo de desenvolvimento das ontologias, tornando-o um processo fácil, rápido e intuitivo. Alguns dos entrevistados destacaram o fato das LPOs auxiliarem na minimização de erros na ontologia, que afeta diretamente também, no tempo de desenvolvimento. Outra questão enfatizada pelos entrevistados é relacionada à definição do escopo. Grande parte dos

entrevistados declarou que o uso de LPOs facilitou o entendimento do domínio e auxiliou na delimitação do escopo do domínio da ontologia. Finalmente, um dos pontos ressaltados pelos entrevistados foi a qualidade final da ontologia resultante desse processo de desenvolvimento. Os participantes declararam que a LPO ajuda a tornar a ontologia bem fundamentada, principalmente por ser baseada em uma ontologia de núcleo.

5.3.3 Discussões

Conforme estabelecido no planejamento do estudo, seu objetivo é avaliar a notação visual das LPOs e analisar o uso de uma LPO na criação de ontologias. Para isso, são considerados os mesmo cinco indicadores utilizados no primeiro experimento, a saber: (i) Grau de dificuldade para entendimento do mecanismo de funcionamento da LPO; (ii) Grau de dificuldade para entendimento dos símbolos da LPO; (iii) Ausência de Problemas de Representação; (iv) Produtividade do desenvolvimento de ontologias; e (v) Qualidade da ontologia resultante.

Com relação ao grau de dificuldade para entendimento do mecanismo de funcionamento da LPO, considerou-se que ele é baixo, uma vez que a maioria dos participantes considerou o mecanismo de fácil entendimento, intuitivo e prático.

Quanto ao grau de dificuldade para entendimento dos símbolos da LPO, concluiu-se ser baixo, mas ainda presente. Os participantes não tiveram dificuldades com a maioria dos símbolos da notação visual, por serem símbolos clássicos de modelos usados tradicionalmente na Engenharia de Software (diagramas de atividades). Alguns poucos elementos geraram dúvidas e, por isso, não foram considerados intuitivos. Vale destacar que a documentação disponibilizada aos participantes ajudou no entendimento desses elementos. Dessa forma, os participantes que tiveram dúvidas relacionadas a esses elementos, entenderam o processo e conseguiram produzir uma ontologia de qualidade em tempo hábil. Ainda que essas dúvidas tenham surgido na aplicação do processo da LPO, os participantes do estudo enfatizaram que, em geral, a notação é clara, intuitiva e de fácil aprendizado.

Embora o grau de dificuldade tenha sido considerado baixo, os participantes destacaram alguns problemas na representação da LPO, dentre eles: dificuldades para identificar caminhos opcionais e obrigatórios, especialmente quando associados a um *fork node*, dificuldade para seguir o fluxo principal de execução do processo, inexistência de indicação de pontos de finalização do processo de aplicação dos padrões.

No que se refere à produtividade do desenvolvimento de ontologias, os participantes declararam que o uso de LPOs aprimorou a produtividade no processo de desenvolvimento

de ontologias de domínio. Os participantes do estudo destacaram não ter sido necessário dispendir muito tempo no processo da criação da ontologia, apesar da grande maioria ter experiência baixa (ou nenhuma) com ontologias e LPOs.

Por fim, em relação à qualidade da ontologia resultante, os participantes também enfatizaram que usar a LPO contribuiu significativamente na qualidade final da ontologia.

Assim, considerando os resultados, pode-se concluir que, no contexto da avaliação realizada, o uso de LPOs contribuiu para o aumento da produtividade do processo de desenvolvimento de ontologias de domínio e para a qualidade das ontologias resultantes, a notação utilizada apresenta baixo grau de dificuldade para entendimento, porém há, ainda, algumas questões relacionadas à representação de LPOs que precisam ser tratadas.

5.3.4 Ameaças à Validade

Todo estudo experimental possui ameaças à sua validade. Elas devem ser tratadas na medida do possível e devem ser consideradas juntamente com os resultados obtidos no estudo. Conforme apresentado no Capítulo 4, elas podem ser classificadas em categorias, a saber: Validade Interna, Validade Externa, Validade de Constructo e Validade de Conclusão. O estudo foi realizado em um contexto muito similar ao do primeiro experimento conduzido neste trabalho, ou seja, os participantes foram alunos do Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo e o estudo foi realizado no contexto de uma disciplina. Dessa forma, as ameaças envolvidas neste estudo são praticamente as mesmas presentes no primeiro estudo experimental realizado, destacando-se uma diferença.

Em relação à Validade Interna, para contornar a ameaça de comunicação e compartilhamento de informações entre participantes, neste estudo foi criado um questionário eletrônico que foi enviado para o e-mail pessoal do participante, de forma que ele pudesse responder individualmente as questões no momento que considerasse mais adequado. Isso minimiza a ameaça da comunicação, uma vez que não é obrigatório que os participantes estejam fisicamente próximos durante a realização do estudo, entretanto, não elimina a possibilidade de que haja comunicação entre eles.

5.4 Considerações Finais

Neste capítulo foram apresentados os primeiros passos realizados para a definição de uma notação visual para LPOs. Foi apresentada uma versão preliminar à notação proposta neste trabalho, a qual foi utilizada no desenvolvimento de S-OPL (FALBO *et al.*, 2016), uma LPO desenvolvida através da extração de padrões da *core ontology* UFO-S (NARDI *et al.*, 2015).

Também foi apresentado um estudo experimental realizado para avaliar a notação utilizada. Os resultados do estudo apontaram alguns problemas na notação visual utilizada, indicando a necessidade de melhorias.

No próximo capítulo é apresentada OPL-ML (*Ontology Pattern Language Modeling Language*), que é resultante de melhorias realizadas na notação visual preliminar, baseadas nos resultados do estudo experimental realizado e também nos resultados do mapeamento sistemático apresentado no Capítulo 3.

Capítulo 6

OPL-ML: Notação Visual para Representação de Linguagens de Padrões Ontológicos

Neste capítulo é apresentada a notação visual proposta neste trabalho. O capítulo encontra-se assim organizado: a Seção 6.1 apresenta uma visão geral sobre a notação visual proposta; a Seção 6.2 apresenta a sintaxe abstrata para LPOs; a Seção 6.3 apresenta a sintaxe concreta para representar LPOs; a Seção 6.4 descreve a execução do processo de sistematização dos princípios de PoN no desenvolvimento da sintaxe concreta; a Seção 6.5 apresenta a prova de conceito da notação proposta, que consistiu na reengenharia de S-OPL; a Seção 6.6 apresenta os resultados de um survey conduzido para avaliar a notação proposta; e na Seção 6.7 são apresentadas as considerações finais do capítulo.

6.1 Introdução

Conforme já discutido neste trabalho, as notações visuais desempenham um papel muito importante na transmissão de informações e são consideradas mais eficazes na comunicação do que o texto. Entretanto, se forem mal projetadas, podem ser bem menos efetivas que texto (MOODY *et al.*, 2010).

Uma notação visual é composta pelos seguintes elementos: um conjunto de símbolos gráficos (vocabulário visual), um conjunto de regras de composição para formação de expressões válidas (gramática visual) e as definições semânticas para cada símbolo (semântica visual). A formação da *sintaxe concreta* é feita pela junção do conjunto de símbolos e das regras de composição. As definições semânticas de cada símbolo, ou seja, os constructos semânticos e seus significados, formam a *sintaxe abstrata*. Os símbolos gráficos são usados para dar significado ou simbolizar os constructos semânticos, geralmente definidos por um metamodelo. Uma expressão na notação visual é chamada de sentença visual ou diagrama. Diagramas são compostos por instâncias dos símbolos gráficos organizados de acordo com as regras da gramática visual (MOODY *et al.*, 2010).

No processo de *design* de uma notação, é importante estar atento tanto à sintaxe abstrata (quais são os constructos semânticos e o que eles significam), quanto à sintaxe concreta (como representar esses constructos). Ambas são importantes para a notação, porém, geralmente, são despendidos maiores esforços na definição da semântica do que na sintaxe visual. Essa falha tende a reduzir a efetividade da notação, dificultando o

entendimento e o uso da notação (MOODY, 2009). Assim, neste trabalho, atenção foi dada ao *design* das duas sintaxes da notação proposta.

Na criação da notação para representação de LPOs proposta neste trabalho e chamada OPL-ML (*Ontology Pattern Language Modeling Language*), optou-se por definir uma sintaxe abstrata e uma sintaxe concreta baseadas em elementos da UML, versão 2.5 (OMG, 2015), que é uma linguagem conhecida no âmbito da modelagem conceitual. Dessa forma, decidiu-se por reutilizar e estender uma parte do metamodelo de UML referente ao diagrama de atividades, o que auxiliou na determinação da sintaxe abstrata e, naturalmente, influenciou na criação da sintaxe concreta, a qual reutiliza diversos símbolos do diagrama de atividades da UML. Essa escolha levou em consideração, ainda, o fato de as LPOs existentes serem representadas por modelos de processo representados usando uma extensão de diagramas de atividade da UML.

Os resultados do mapeamento sistemático apresentado no Capítulo 3 também foram usados para apoiar a definição de OPL-ML, bem como de mecanismos de apoio necessários. Buscando-se tratar a lacuna “ausência de uma visão completa das LPs, que apresente aspectos relacionados ao processo da LP e à sua estrutura”, decidiu-se incluir em OPL-ML modelos distintos para tratar as perspectivas comportamental e estrutural de uma LPO. Assim, a notação visual definida neste trabalho propõe a utilização de um modelo para representar o processo da LPO e outro para representar aspectos estruturais da LPO. Além disso, para tratar a lacuna “falta de preocupação em prover mecanismos para auxiliar a seleção dos padrões”, o modelo comportamental gerado na notação visual para processo da LPO serve de guia para a seleção dos padrões.

Por fim, buscando-se obter uma linguagem cognitivamente rica e suprir a lacuna “as notações visuais adotadas são cognitivamente pobres”, durante o *design* da sintaxe concreta da notação visual proposta, foram considerados princípios de PoN (MOODY, 2009). O uso desses princípios foi guiado pelo processo de aplicação de PoN proposto em (TEIXEIRA *et al.*, 2016). Nas próximas seções OPL-ML é apresentada, bem como a aplicação dos princípios de PoN durante seu desenvolvimento.

6.2 Sintaxe Abstrata

Uma sintaxe abstrata define quais elementos são necessários para expressar o objetivo da notação. É nela que se encontra cada constructo semântico e seu significado. A Figura 5.1 apresenta o metamodelo geral da sintaxe abstrata de OPL-ML. Optou-se por apresentar os metamodelos em inglês para manter consistência com o metamodelo de UML.

Conforme ilustrado na Figura 6.1, uma OPL (*Ontology Pattern Language*) é composta por um processo (*OPLProcess*) e por um modelo estrutural (*OPLStructuralModel*).

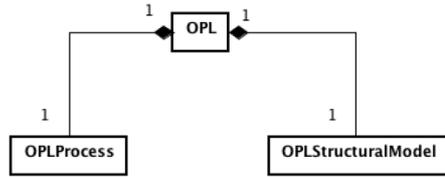


Figura 6.1 – Metamodelo Geral de LPOs

A Figura 6.2 apresenta o metamodelo referente à perspectiva estrutural de LPOs.

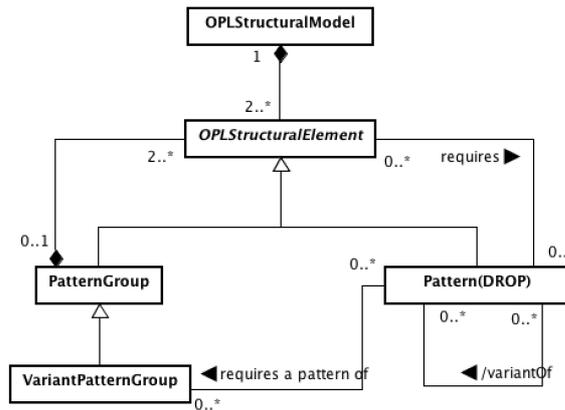


Figura 6.2 - Metamodelo da visão estrutural de LPOs.

O modelo estrutural de LPOs (*OPLStructuralModel*) é composto por elementos estruturais de LPOs (*OPLStructuralElement*), que podem ser padrões ontológicos de domínio (*Pattern(DROP)*) ou grupos de padrões (*PatternGroup*). Um grupo de padrões (*PatternGroup*) pode ser composto por outros por elementos estruturais de LPO (*OPLStructuralElement*), ou seja, por padrões (*Pattern(DROP)*) ou outros grupos de padrões (*PatternGroup*). Se um elemento estrutural de um modelo estrutural de LPO é parte de um grupo de padrões, então o elemento estrutural é parte do modelo estrutural da LPO. Dessa forma,

$$\forall sm: OPLStructuralModel, se: OPLStructuralElement, pg: PatternGroup \\ partOf(se, pg) \wedge partOf(pg, sm) \rightarrow partOf(se, sm).$$

Um grupo de padrões variantes (*VariantPatternGroup*) é um grupo de padrões (*PatternGroup*) composto por padrões (*Pattern(DROP)*) que tratam um mesmo problema e cuja aplicação é mutuamente exclusiva. Dessa forma, se dois padrões fazem parte de um grupo de padrões variantes, um padrão é variante do outro. Assim,

$$\forall p1, p2: Pattern(DROP), vpg: VariantPatternGrupo \\ partOf(p1, vpg) \wedge partOf(p2, vpg) \rightarrow variantOf(p1, p2).$$

A relação *variantOf* é uma relação simétrica. Assim, se um padrão *p1* é variante de um padrão *p2*, então *p2* é variante de *p1*. Logo,

$$\forall p1, p2: Pattern(DROP) \text{ variantOf}(p1, p2) \rightarrow \text{variantOf}(p2, p1).$$

Padrões podem ser requeridos por outros padrões ou por grupos de padrões (*OPLStructuralElement requires Pattern(DROP)*). A relação *requires* implica que há uma relação de dependência entre esses padrões. Padrões que são parte de um grupo de padrões que requer um dado padrão também requerem esse padrão. Assim,

$$\forall p1, p2: Pattern(DROP), pg: PatternGroup \text{ partOf}(p1, pg) \wedge \text{requires}(pg, p2) \rightarrow \text{requires}(p1, p2).$$

Um padrão pode requerer um padrão de um grupo de padrões variantes (*Pattern(DROP) requires a pattern of VariantPatternGroup*). Uma vez que os padrões que formam um grupo variante são mutuamente exclusivos, a relação entre um padrão e um grupo variante refere-se à necessidade de que um dos padrões que fazem parte do grupo de padrões variante seja aplicado antes do padrão que o requer.

A Figura 6.3 apresenta o metamodelo referente à perspectiva comportamental (processo) de LPOs. O metamodelo é uma extensão de um fragmento do metamodelo que trata aspectos do diagrama de atividades da UML. Ele inclui os constructos semânticos necessários para a definição de um processo de LPO. Na figura, os conceitos oriundos do metamodelo da UML encontram-se em branco e os inclusos para tratar a sintaxe abstrata de processo de LPOs são destacados em amarelo.

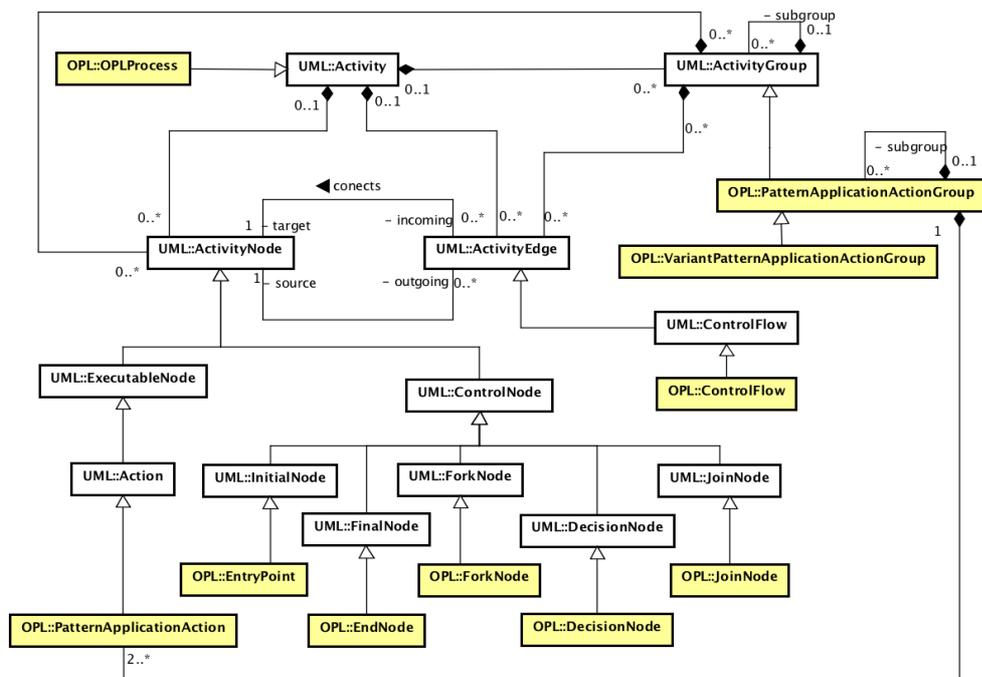


Figura 6.3 - Metamodelo da visão comportamental de LPOs.

Uma atividade (*Activity*) é um comportamento especificado como uma sequência de unidades subordinadas utilizando-se modelos de fluxos de controle. Comportamentos subordinados coordenados por esses modelos podem ser iniciados pelo término da execução de outros comportamentos no modelo ou por eventos que ocorrem externamente ao fluxo (OMG, 2015).

O fluxo de execução é modelado como nós de atividade (*ActivityNodes*) conectados por arestas de atividade (*ActivityEdges*). Nós de atividade (*ActivityNodes*) são usados para modelar passos individuais no comportamento especificado por uma atividade (*Activity*). Aresta de atividade (*ActivityEdge*), por sua vez, é uma conexão direcionada entre dois nós de atividade (*ActivityNodes*). Fluxos de controle (*ControlFlows*) são arestas de atividade (*ActivityEdges*) usados para explicitamente sequenciar a execução de nós de atividade (*ActivityNodes*). Fluxos de controle (*ControlFlows*) são direcionados do nó fonte para o nó alvo e são fundamentais controlar as possíveis rotas do fluxo de execução (OMG, 2015).

Um nó de atividade (*ActivityNode*) pode ser um nó executável (*ExecutableNode*), que diz respeito à execução de um comportamento subordinado, ou um nó de controle (*ControlNode*), que trata aspectos como sincronização, decisão e concorrência. Ações (*Actions*) são um tipo concreto de nó executável (*ExecutableNode*). Uma ação (*Action*) é uma unidade fundamental de uma funcionalidade executável contida, direta ou indiretamente, em um comportamento. Nós de atividade (*ActivityNodes*) e arestas de atividade (*ActivityEdges*) podem ser agrupados em grupos de atividades (*ActivityGroups*) (OMG, 2015).

Processo de LPO (*OPLProcess*) é uma especialização de atividade (*Activity*) na qual o comportamento especificado diz respeito à aplicação de padrões ontológicos. Ações de aplicação de padrão (*PatternApplicationActions*) são os nós executáveis (*ExecutableNodes*) usados na modelagem do fluxo de execução de um processo de LPO (*OPLProcess*), juntamente com nós de controle (*ControlNodes*) e fluxos de controle (*ControlFlows*). Uma ação de aplicação de padrão (*PatternApplicationAction*), como o próprio nome sugere, é uma ação (*Action*) referente à aplicação de um padrão.

Os nós de controle (*ControlNodes*) que podem ser usados na modelagem do fluxo de execução de um Processo de LPO (*OPLProcess*) são: nó inicial (*InitialNode*), nó final (*EndNode*), nó de bifurcação (*ForkNode*), nó de decisão (*DecisionNode*) e nó de junção (*JoinNode*). Esses constructos são especializações de constructos da UML (como mostra a Figura 5.3) e são descritos a seguir:

- Ponto de Entrada (*Entry Point*): age como o ponto de partida para a execução de um processo de LPO (*OPLProcess*). É uma especialização do constructo nó inicial

(*InitialNode*) da UML e apresenta uma importante diferença em relação a esse constructo. Em um diagrama de atividades UML, quando há mais de um nó inicial (*InicialNode*) representado, a execução da atividade (*Activity*) tem início com a execução de múltiplos fluxos concorrentes, um para cada nó inicial. Entretanto, em um processo de LPO (*OPLProcess*), a existência de mais de um ponto de entrada (*EntryPoint*) indica que apenas um seja selecionado para dar início ao fluxo de execução do processo de LPO (*OPLProcess*).

- Ponto Final (*EndPoint*): é uma especialização do constructo nó final (*FinalNode*) da UML e mantém sua semântica, ou seja, representa o término do fluxo de execução do processo de LPO (*OPLProcess*).
- Nó de Bifurcação (*Fork Node*): é uma especialização do constructo de mesmo nome na UML e mantém sua semântica. Um nó de bifurcação (*ForkNode*) divide o fluxo de execução em múltiplos fluxos concorrentes. Ele tem exatamente um fluxo de controle de entrada e múltiplos fluxos de controle de saída, sendo que todos os fluxos de controle de saída devem ser seguidos durante o fluxo de execução do processo de LPO (*OPLProcess*).
- Nó de Junção (*JoinNode*): é uma especialização do constructo de mesmo nome na UML e mantém sua semântica. Sincroniza múltiplos fluxos concorrentes. Ele tem diversos fluxos de controle de entrada e apenas um fluxo de controle de saída, sendo que todos os fluxos de controle de entrada devem ser seguidos durante o fluxo de execução do processo de LPO (*OPLProcess*).
- Nó de Decisão (*DecisionNode*): é uma especialização do constructo de mesmo nome na UML e mantém sua semântica. Apresenta opções de fluxos de controle no fluxo de execução do processo. Tem um fluxo de controle de entrada e múltiplos fluxos de controle de saída, sendo que apenas um dos fluxos de saída pode ser selecionado para dar continuidade ao fluxo de execução do processo. Geralmente, há condições de guarda que auxiliam na escolha do fluxo a ser seguido.

Grupos de atividades (*ActivityGroups*) que agrupam constructos relacionados à OPL (i.e., ações de aplicação de padrão (*PatternApplicationActions*), fluxos de controle (*ControlFlows*) e os nós de controle (*ControlNodes*) apresentados anteriormente) são ditos grupos de ações de aplicação de padrões (*PatternApplicationActionGroups*). Esses grupos podem, ainda, agrupar outros grupos de ações de aplicação de padrões (*PatternApplicationActionGroups*), ditos seus subgrupos (*subgroup*). Nesse contexto, se uma ação

de aplicação de padrão faz parte de um subgrupo, então a ação também faz parte do grupo que contém o referido subgrupo. Assim,

$$\forall paa: PatternApplicationAction, sg: subgroup, paag: PatternApplicationActionGroup \\ partOf(paa, sg) \wedge partOf(sg, paag) \rightarrow partOf(paa, paag).$$

Quando um Grupo de Ações de Aplicação de Padrões (*PatternApplicationActionGroup*) é formado por Ações de Aplicação de Padrões (*PatternApplicationAction*) que são mutuamente exclusivas, ou seja, que são aplicações de padrões variantes, tem-se um Grupo de Ações de Aplicação de Padrões Variantes (*VariantPatternApplicationActionGroup*).

A Tabela 6.1 sumariza os constructos da sintaxe abstrata referente às perspectivas estruturais (*LPOStructuralModel*) e de comportamento (*LPOProcess*) de uma LPO.

Tabela 6.1 - Constructos semânticos da sintaxe abstrata das visões estrutural e de processo de LPOs.

Perspectiva Estrutural (<i>LPOStructuralModel</i>)
Padrão
Grupo de Padrões
Grupo de Padrões Variantes
Relação <i>requires</i>
Relação <i>requires a pattern of</i>
Perspectiva Comportamental (<i>LPOProcess</i>)
Ação de Aplicação de Padrão
Grupo de Ações de Aplicação de Padrões
Grupo de Ações de Aplicação de Padrões Variantes
Fluxo de Controle
Ponto de Entrada
Ponto Final
Nó de Bifurcação
Nó de Junção
Nó de Decisão

6.3 Sintaxe Concreta

Após a definição dos constructos semânticos da notação visual (i.e., a sintaxe abstrata), devem ser definidos os símbolos para representar os constructos. O conjunto de símbolos forma a sintaxe concreta da notação visual.

Uma vez que o metamodelo adotado para definir os constructos referentes ao processo de uma LPO é uma extensão de parte do metamodelo para diagrama de atividades da UML, sempre que possível adotou-se para representar os constructos a mesma sintaxe concreta usada no diagrama de atividades da UML, a fim de favorecer o uso por usuários que já conheçam essa notação. Além disso, durante a definição da sintaxe concreta, foram considerados princípios de PoN para a obtenção de símbolos cognitivamente viáveis. A

Tabela 6.2 apresenta os símbolos definidos para cada constructo semântico do modelo estrutural de OPL-ML e a Tabela 6.3 apresenta os símbolos definidos para cada constructo semântico do modelo comportamental de OPL-ML. Na próxima seção, o uso dos princípios de PoN durante o *design* da sintaxe concreta é discutido.

Tabela 6.2 - Elementos da sintaxe concreta da notação visual proposta: Modelo Estrutural.

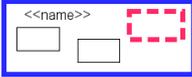
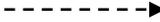
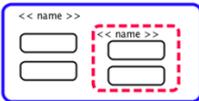
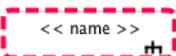
Perspectiva Estrutural (<i>LPOStructuralModel</i>)		
Elemento	Representação gráfica	Descrição
Padrão		Retângulo com rótulo
Grupo de Padrões (no formato expandido)		Região limitada por linhas retas com rótulo. Sugere-se o uso da cor azul nas linhas sempre que for possível.
Grupo de Padrões (no formato caixa preta)		Retângulo com o símbolo usado em alguns constructos da UML na parte inferior à direita e com rótulo.
Grupo de Padrões Variantes (no formato expandido)		Região limitada por linhas retas tracejadas e com rótulo. Sugere-se o uso da cor vermelha nas bordas sempre que for possível.
Grupo de Padrões Variantes com Gerência de complexidade (no formato caixa-preta)		Região limitada por linhas retas tracejadas e com o símbolo usado em alguns constructos da UML na parte inferior à direita e rótulo. Sugere-se o uso da cor vermelha nas bordas sempre que for possível.
Relação requires		Seta direcionada sólida
Relação requires a pattern of		Seta direcionada tracejada

Tabela 6.3 - Elementos da sintaxe concreta da notação visual proposta: Modelo Comportamental.

Perspectiva Comportamental (<i>LPOProcess</i>)		
Elemento	Representação gráfica	Descrição
Ponto de Entrada		Círculo preenchido
Ponto Final		Círculo preenchido com borda dupla
Ação de Aplicação de Padrão		Retângulo com bordas arredondadas e rótulo
Fluxo de Controle		Seta direcionada
Nó de Decisão		Losango
Nó de Bifurcação (<i>Fork Node</i>)		Barra preta com um fluxo de entrada e vários de saída.
Nó de Junção (<i>Join Node</i>)		Barra preta com vários fluxos de entrada e apenas um de saída.
Grupo de Ações de Aplicação de Padrões (no formato expandido)		Região limitada por linhas retas com cantos arredondados e com rótulo. Sugere-se usar a cor azul nas linhas sempre que for possível.
Grupo de Ações de Aplicação de Padrão (no formato caixa-preta)		Retângulo com bordas arredondadas e o símbolo usado em alguns constructos da UML na parte inferior à direita do retângulo e com rótulo. Sugere-se usar a cor azul nas linhas sempre que for possível.
Grupo de Ações de Aplicação de Padrões Variantes (no formato expandido)		Região limitada por linhas retas tracejadas com cantos arredondados e com o símbolo usado em alguns constructos da UML na parte inferior à direita e rótulo. Sugere-se usar a cor vermelha nas linhas sempre que for possível.
Grupo de Ações de Aplicação de Padrões Variantes (no formato caixa-preta)		Retângulo com linhas tracejadas e bordas arredondadas e com o símbolo usado em alguns constructos da UML na parte inferior à direita e rótulo. Sugere-se usar a cor vermelha nas linhas sempre que for possível.

Além do conjunto de símbolos apresentados, algumas orientações devem ser consideradas para um efetivo uso da sintaxe concreta:

- i. Deve-se usar o mesmo nome para representar elementos equivalentes nos modelos estrutural e de processo. Por exemplo, um Padrão no modelo estrutural possui uma Ação de Aplicação de Padrão correspondente no modelo de processo, devendo esses elementos serem identificados pelo menos nome.
- ii. Caso opte-se por adotar cores diferentes em um diagrama, deve-se usar a mesma cor para representar os padrões de um mesmo grupo.
- iii. Caso opte-se por adotar cores diferentes, deve-se usar a mesma cor para representar elementos equivalentes nos modelos estrutural e de processo. Por exemplo, se os Padrões de um grupo no modelo estrutural são preenchidos pela cor amarela, as Ações de Aplicação de Padrão equivalentes no modelo de processo também deverão ser preenchidas em amarelo.

6.4 Aplicação dos Princípios de PoN no *Design* da Sintaxe Concreta

Um dos trabalhos mais conhecidos sobre *design* de notações visuais cognitivamente eficazes é a Física das Notações (*Physics of Notations – PoN*), apresentada em (MOODY, 2009). Conforme discutido no Capítulo 2, PoN baseia-se em nove princípios: Clareza Semiótica, Discriminabilidade Perceptiva, Transparência Semântica, Gestão de Complexidade, Integração Cognitiva, Expressividade Visual, Codificação Dupla, Economia Gráfica e Ajuste Cognitivo. Essa teoria pode ser usada para comparar, avaliar, aprimorar e construir notações visuais.

Neste trabalho, realizou-se o *design* da sintaxe concreta de OPL-ML (apresentada na seção anterior) considerando-se os princípios de PoN. Para tal, utilizou-se um processo de sistematização da aplicação dos princípios de PoN proposto por (TEIXEIRA *et al.*, 2016) e apresentado no Capítulo 2. Esse processo define passos para o *design* da sintaxe concreta de uma linguagem gráfica, de modo a guiar a aplicação dos princípios de PoN no objetivo pretendido.

O primeiro passo do processo, ***Definir Conjunto de Dialeto***, é baseado no princípio de Ajuste Cognitivo, que trata da necessidade de prover diferentes dialetos para diferentes tarefas ou classes de usuários da linguagem. Para definir a quantidade de dialetos necessários para uma certa notação visual, o processo requer que sejam analisados: o tipo do

domínio do problema, a sintaxe abstrata, o perfil dos *stakeholders* e as tarefas de modelagem que utilizarão a notação visual. A análise desses aspectos é apresentada a seguir:

- Tipo do Domínio do Problema: A notação visual proposta deve ser capaz de representar o processo de aplicação de padrões ontológicos de LPOs e as relações de dependência entre os padrões.
- Sintaxe Abstrata: Os constructos semânticos que requerem símbolos (sintaxe concreta) são aqueles apresentados na Tabela 6.1.
- Perfil dos Stakeholders: Os usuários de LPOs são tipicamente Engenheiros de Ontologias (iniciantes ou experientes) que usarão LPOs para o desenvolvimento de ontologias de domínio.
- Tarefas de Modelagem: LPOs são tipicamente usadas para o desenvolvimento de ontologias de domínio.

Apesar de haver *stakeholders* com níveis de experiência diferentes, a sintaxe concreta utilizada deve ser simples e intuitiva para todos os *stakeholders*. Dessa forma, analisando-se a necessidade de um conjunto de dialetos, concluiu-se não haver necessidade de mais de um dialeto e decidiu-se pelo uso de um único dialeto que pode ser caracterizado a partir da seguinte descrição: “Uma notação visual simples, objetiva e intuitiva para usuários com alguma experiência mínima em modelagem conceitual. É necessário que contenha os símbolos suficientes para representar modelos de processo e estrutural de LPOs, de modo a não haver ambiguidades. Além disso, é importante que, em caso do uso de cores, seja possível transpassar a notação visual para escala de cinza, sem prejudicar o entendimento do processo”.

Tendo-se caracterizado o dialeto, o próximo passo foi **Definir o Conjunto de Símbolos do Dialeto**, que resultou nos elementos apresentados anteriormente nas Tabelas 6.2 e 6.3. No âmbito desse passo foram considerados os princípios de Clareza Semiótica, Transparência Semântica e Discriminabilidade Perceptiva bem como Expressividade Visual, Economia Gráfica e Codificação Dupla, considerados princípios de apoio no processo definido em (TEIXEIRA *et al.*, 2016). A seguir são apresentadas discussões sobre esses princípios no contexto da sintaxe concreta proposta.

Clareza Semiótica

Em relação a Clareza Semiótica, nota-se que quase todos os elementos possuem uma correspondência 1:1 entre o constructo do metamodelo e o símbolo gráfico.

Em relação ao modelo de processo (referente a *OPLProcess*), apesar dos símbolos de *ForkNode* e *JoinNode* serem similares, eles se diferem pelo número de fluxos de entrada e saída definidos. Considerando-se os símbolos definidos para o modelo de processo e para o modelo estrutural (referente a *OPLStructural Model*), em ambos os casos os agrupamentos são regiões delimitadas por linhas retas. Apesar dessa característica em comum, cada agrupamento possui uma variável visual que o difere dos outros. Grupo de Ações de Aplicação de Padrão e Grupo de Ações de Aplicação de Padrões Variantes têm em comum as bordas arredondadas, mas se diferem na textura da linha. O último possui linhas tracejadas, enquanto o primeiro é definido com linhas sólidas. Essa mesma diferenciação por textura da linha ocorre com os elementos de agrupamentos do modelo estrutural. É importante notar, também, que os elementos de agrupamentos do modelo de processo se diferem dos elementos de agrupamentos do modelo estrutural pelo tipo de junção das linhas retas. No modelo de processo, essa junção é feita com bordas arredondadas, enquanto que no modelo estrutural forma um ângulo de 90 graus.

Vale destacar também, que apesar de os símbolos definidos para os grupos em formato não expandido possuírem certa semelhança com os seus respectivos símbolos de grupos expandidos, eles possuem características diferentes. Os símbolos dos grupos expandidos são regiões delimitadas por linhas retas, que podem gerar formas diversas. Já os símbolos dos grupos não expandidos são definidos como um retângulo (com bordas retas nos modelos estruturais e arredondadas nos modelos de processo) com o ícone da UML para representar atividades complexas, detalhadas em outro diagrama, no canto inferior direito.

Há duas exceções nesse mapeamento isomórfico. Primeiro, o constructo *OPLStructuralElement* é um elemento de modelagem abstrato e, portanto, não é necessário definir um símbolo para eles. Segundo, os relacionamentos *variantOf* existentes em ambos os modelos são associações derivadas, que também dispensam a necessidade de um símbolo específico.

Por fim, não foram definidos símbolos específicos para representar os elementos *OPLProcess* e *OPLStructuralModel*, uma vez que esses elementos representam os modelos como um todo e não são representados graficamente nos diagramas.

Transparência Semântica

A sintaxe concreta definida para os modelos de processo e estrutural de LPOs é baseada em formas geométricas. Dessa forma, pode-se concluir que os símbolos escolhidos são considerados semanticamente opacos, pois não sugerem o seu significado de uma forma

direta. Entretanto, a transparência semântica também pode ser analisada através do quão facilmente lembrado é um símbolo. Desse modo, considerando que os símbolos escolhidos para o modelo de processo são basicamente os mesmos de diagramas de atividades da UML (a exceção fica por conta dos símbolos referentes aos agrupamentos), uma notação bastante usada para representar processos, optou-se por mantê-los na sintaxe concreta, objetivando favorecer a aprendizagem desses elementos. O mesmo acontece com o modelo estrutural, que apesar de não ser baseado em um modelo específico, possui símbolos comumente usados em linguagens de padrões, tais como o retângulo para representar padrões e as setas para representar as relações. Esses símbolos foram encontrados em muitas das LPs analisadas no mapeamento sistemático apresentado no Capítulo 3.

Dentre os símbolos da sintaxe concreta, pode-se considerar que a seta usada para representar os fluxos é a que apresenta maior transparência semântica. Os símbolos escolhidos para representar os grupos em seu formato expandido em ambos os modelos também atendem fortemente a esse princípio, pois sugerem agrupamento.

Discriminabilidade Perceptiva

Analisando-se a sintaxe concreta para os constructos do modelo de processo, observa-se que os símbolos definidos para Grupo de Ações de Aplicação de Padrão e Grupo de Ações de Aplicação de Padrões Variantes possuem uma pequena distância visual, pois ambos são regiões delimitadas por linhas retas com cantos arredondados. Porém, utilizou-se da variável visual textura para diferenciá-los. A linha do símbolo para Grupo de Ações de Aplicação de Padrões Variantes é tracejada, enquanto a do Grupo de Ações de Aplicação de Padrão é uma linha sólida. Pode-se observar que o mesmo ocorre com os agrupamentos do modelo estrutural. A diferença dos agrupamentos estruturais em relação aos de processo se dá pelo tipo das junções das linhas retas, que são arredondadas no modelo de processo, enquanto que no modelo estrutural formam um ângulo de 90°.

A forma dos símbolos de Ponto de Entrada e Ponto Final são bem similares. A variável visual utilizada para distingui-los foi a textura. Há uma linha que circunda o círculo do símbolo de Ponto Final, a qual proporciona a diferenciação necessária para distingui-lo do Ponto de Entrada.

É perceptiva, também, a pequena distância visual entre os elementos Nó de Bifurcação e Nó de Junção. A diferença na forma se dá pela quantidade de entradas e saídas. No Nó de Bifurcação há um único fluxo de controle de entrada e vários de saída. Um Nó de Junção, por sua vez, apresenta mais de um fluxo de controle de entrada e apenas um de saída.

Essa pequena distância visual é aceita neste princípio devido à pequena distância semântica existente entre eles, o que acarreta em uma consistência entre a distância visual e a distância semântica desses elementos.

Os outros elementos da sintaxe concreta se diferem principalmente pela forma.

Expressividade Visual

Este princípio é fortemente relacionado à Discriminabilidade Perceptiva e ressalta a importância de os símbolos serem visualmente expressivos. Para isso, deve-se explorar o máximo de variáveis visuais possíveis na definição da sintaxe concreta.

Observando-se a sintaxe concreta como um todo, é possível concluir que ficou-se limitado ao uso das seguintes variáveis visuais: forma, textura e tamanho. A cor é uma variável visual que é utilizada como codificação redundante, porque as diferenças desaparecem quando os diagramas são impressos em escala de cinza. As demais variáveis visuais não foram exploradas devido à decisão de se adotar uma sintaxe concreta similar a uma já amplamente conhecida na Engenharia de Software e usada em diagramas de atividades da UML.

Economia Gráfica

A sintaxe concreta do modelo de processo para LPOs é composta por nove elementos e a do modelo estrutural por sete elementos. PoN defende o uso de até seis elementos em um dialeto. Porém, na caracterização do dialeto definiu-se ser importante que o dialeto proposto contenha símbolos suficientes para representar, sem ambiguidades, processos e modelos estruturais de LPOs. Portanto, dá-se prioridade aos princípios Discriminabilidade Perceptiva e Expressividade Visual sobre o princípio de Economia Gráfica.

Codificação Dupla

Este princípio lida com o texto como um complemento de informação do símbolo gráfico. Na sintaxe concreta proposta, texto não foi utilizado como complemento de informação. Contudo, vale ressaltar que o texto tem um papel fundamental no nível de instância (diagramas construídos usando a notação proposta). Os rótulos (*labels*) existentes nos símbolos são parte integrante deles e, portanto, devem ser providos visando apoiar a definição e interpretação do significado pretendido para os símbolos que requerem rótulos.

Utiliza-se, opcionalmente, a codificação redundante por meio de cores das linhas dos Grupos de Ações de Aplicação de Padrão (azul) e Grupos de Ações de Aplicação de Padrões Variantes (vermelho), no modelo de processo; e, Grupos de Padrões (azul) e Grupos de

Padrões Variantes (vermelho), no modelo estrutural. Além disso, cores podem ser usadas no nível de instância para preencher grupos em formato não expandido, ações de aplicação de padrão e padrões. No modelo de processo, a recomendação é que se utilize a mesma cor de preenchimento para um Grupo de Ações de Aplicação de Padrão e as Ações de Aplicação de Padrão que dele fazem parte. Do mesmo modo, no modelo estrutural, recomenda-se utilizar a mesma cor de preenchimento para um Grupo de Padrões e os Padrões que dele fazem parte, mantendo-se as mesmas cores nos correspondentes modelos estrutural e de processo.

Após a definição do conjunto de elementos da sintaxe concreta, realizou-se o passo ***Identificar Formas de Gerenciar a Complexidade dos Modelos***, onde foi feita a análise da organização e interação dos elementos segundo os princípios de Gestão de Complexidade e Integração Cognitiva, conforme apresentado a seguir.

Gestão de Complexidade

Como informado no Capítulo 2, este princípio destaca a importância de gerenciar a complexidade diagramática. Ela é medida pela quantidade de elementos em um diagrama. Notou-se a necessidade de gerenciar a complexidade, uma vez que deve ser possível representar LPOs complexas, contendo muitos elementos, o que pode dificultar o entendimento do diagrama resultante. De modo a incrementar a rapidez e precisão do entendimento dos diagramas, optou-se por fazer uma redução do montante de informação através do uso de modularização, permitindo, assim, que haja um diagrama que proveja uma visão do processo e das relações estruturais como um todo. Para tanto, utilizou-se técnicas de contextualização (foco + conteúdo), de modo que apenas a parte de interesse corrente (foco) tenha seu conteúdo apresentado.

Para a representação dos módulos, foram utilizados os símbolos dos agrupamentos como caixa-preta, de modo a ocultar os padrões ou ações de padrões internos a eles. Nas representações, adicionou-se, ainda, o ícone da UML para representar atividades complexas no canto inferior direito para indicar que o conteúdo do agrupamento está sendo omitido. Desse modo, podem-se construir modelos com níveis de abstração diferentes: um mais abstrato, mostrando apenas as relações entre os grupos, e outro mais detalhado, mostrando os grupos e também as relações entre grupos e os elementos internos a eles. Esses recursos possibilitam maior clareza no diagrama e entendimento eficaz do processo e da estrutura da LPO como um todo.

Integração Cognitiva

O princípio da integração cognitiva define que toda notação deve ter mecanismos que permitam integrar informações de diferentes diagramas. Os símbolos escolhidos para gerenciar a complexidade diagramática são conhecidos por agrupar elementos e ter informações internas, deixando claro o significado do símbolo no diagrama, facilitando assim o aprendizado do modelo, de modo a permitir um fácil rastreamento de informações entre os diagramas.

Outro mecanismo utilizado para integrar os diagramas é a utilização de rótulos e cores iguais para os Padrões e as Ações de Aplicação de Padrão relacionados. Para cada Padrão do modelo estrutural existe uma Ação de Aplicação de Padrão correspondente no modelo de processo da LPO. E ambos devem possuir o mesmo nome e a mesma cor. De forma similar, os grupos e outros elementos dos modelos da LPO são relacionados. Explicitar a conexão entre os elementos dos dois modelos por meio de cores e nomes iguais auxilia o usuário no entendimento e uso da OPL.

6.5 Reengenharia da Representação Visual de uma LPO

Para verificar a viabilidade de utilização de OPL-ML, foi realizada como prova de conceito a reengenharia da representação visual de S-OPL - *Service Ontology Pattern Language* (FALBO *et al.*, 2016). Segundo (OATES, 2006), uma prova de conceito é uma avaliação que mostra que uma proposta é exequível, porém não é suficiente para afirmar se ela funciona em um contexto real e requer avaliações complementares.

A seguir são discutidas as principais alterações realizadas em S-OPL e sua nova representação é apresentada.

(i) Separação de Aspectos Estruturais e Comportamentais

Conforme discutido na Seção 6.1, não é recomendável representar em um mesmo diagrama as perspectivas comportamental e estrutural de uma LPO. Assim, foi criado o Modelo Estrutural de S-OPL, que descreve os elementos que a compõem e as relações estruturais entre eles. A Figura 6.4 apresenta esse modelo.

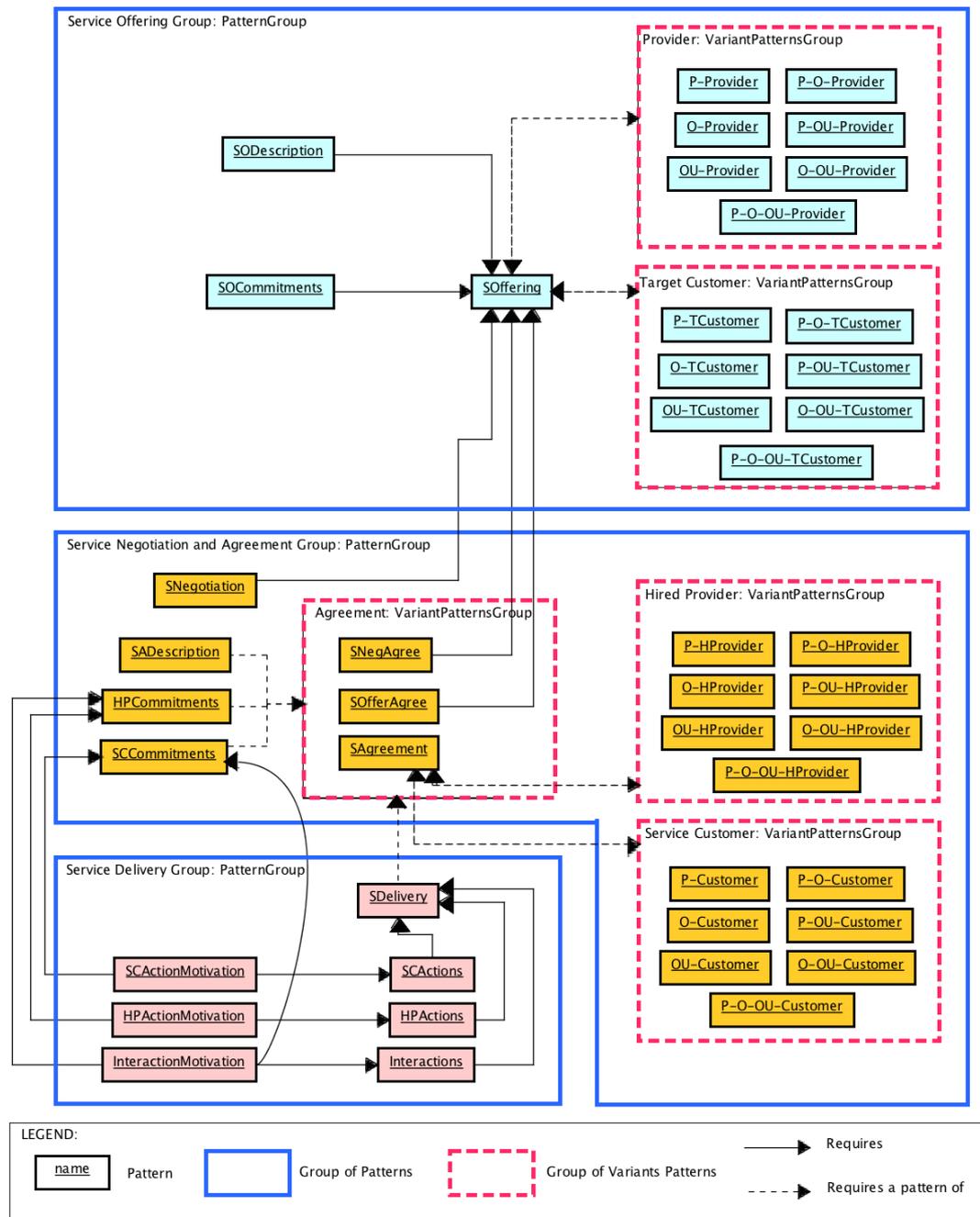


Figura 6.4 - Modelo Estrutural de S-OPL (formato expandido).

(ii) Modelo de processo com fluxo principal obrigatório

Na notação visual inicialmente adotada para representar S-OPL, o modelo de processo era descrito através de fluxos opcionais. O engenheiro de ontologias podia parar de modelar a ontologia em qualquer parte do modelo de processo. Porém, esse comportamento dos fluxos, além de não estar em linha com a UML, proporcionava muitos problemas de interpretação. Em OPL-ML introduziu-se a noção de processo a ser seguido passo a passo, de um ponto de entrada até um ponto final. Dessa forma, surgiu um fluxo

principal a ser seguido e, de acordo com as decisões a serem tomadas pelo engenheiro de ontologias, a ontologia é modelada até encontrar um ponto final no processo. Essa mudança no fluxo de controle do modelo de processo implicou na adição de novos nós de controle e na alteração na semântica de outros.

(iii) Uso do Nó de Finalização (End Node)

A notação visual usada na versão inicial de S-OPL não possuía um elemento para indicar a finalização do processo, uma vez que os caminhos eram definidos como opcionais (quando obrigatórios, eles deveriam ser estereotipados com <<mandatory>>). Assim, na versão inicial de S-OPL, os pontos do processo onde o engenheiro de ontologias poderia encerrar a aplicação dos padrões não eram definidos, podendo o engenheiro interromper o processo em qualquer ponto, desde que o próximo elemento não fosse um fluxo obrigatório. Na nova versão de S-OPL nós de finalização foram inseridos indicando quando é possível ao engenheiro de ontologias finalizar a execução do processo. A Figura 6.5 mostra um exemplo dessa alteração.

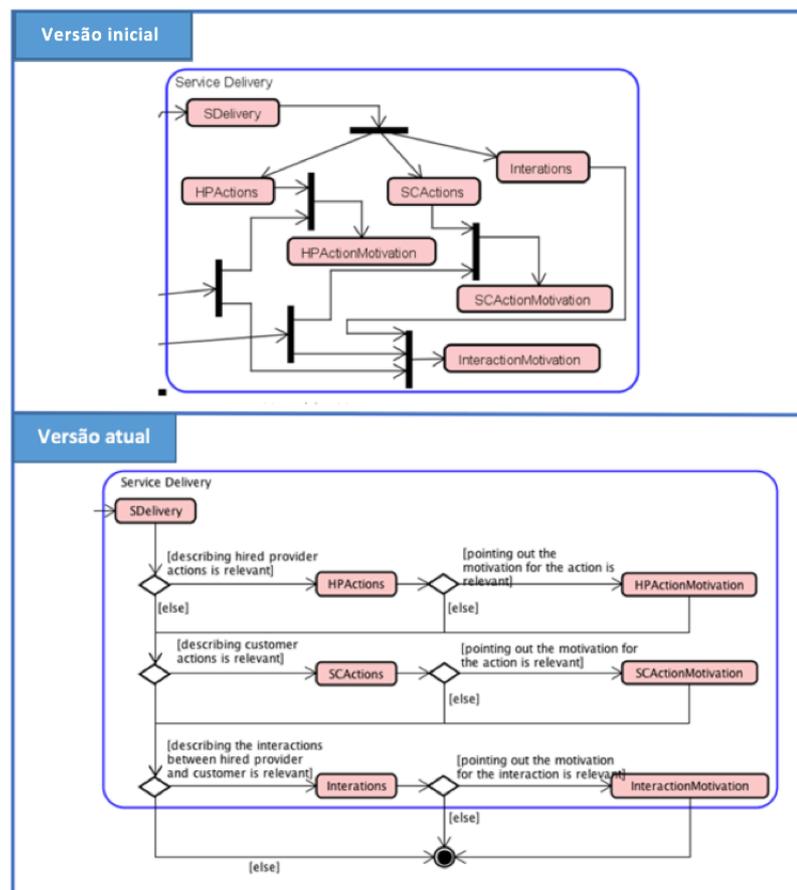


Figura 6.5 - Adição de End Points.

(iv) Uso do Nó de Decisão (*Decision Node*)

Com a alteração na semântica do fluxo de controle, nós de decisão passam a ter de ser usados em todas as situações onde há caminhos opcionais. Essa alteração também contribui para evitar dúvidas sobre os caminhos a serem seguidos e a obrigatoriedade ou não de segui-los. No nó de decisão um (e apenas um) caminho deve ser seguido no fluxo de execução do processo. A Figura 6.6 ilustra um exemplo dessa alteração.

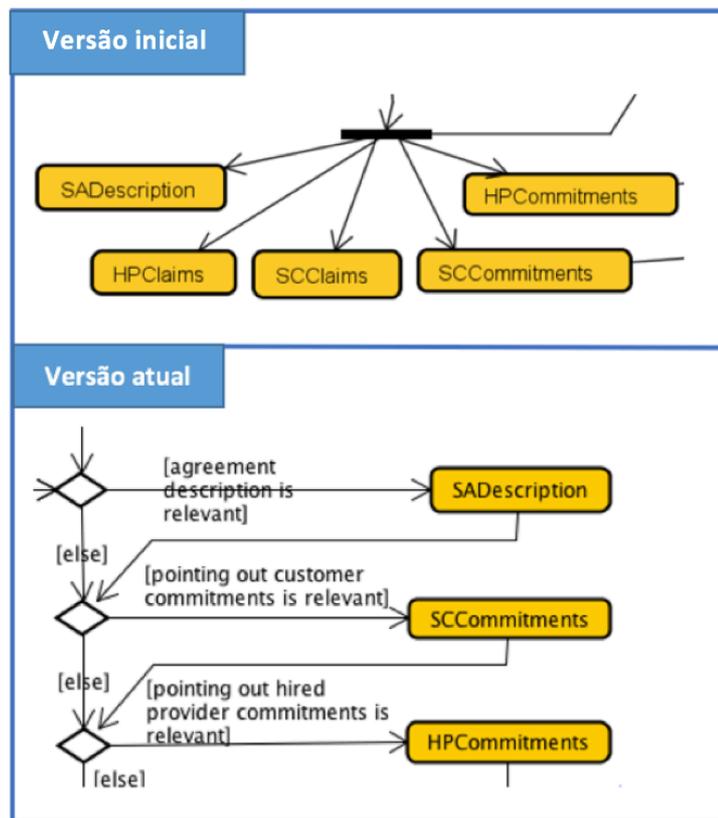


Figura 6.6 - Utilização de Decision Nodes.

(v) Uso do Nó de Bifurcação (*Fork Node*)

Na versão inicial de S-OPL, as saídas do nó de bifurcação são caminhos opcionais. A nova notação proposta adota esse elemento da mesma forma que é usado em diagramas de atividades da UML. Assim, as saídas do nó de bifurcação passam a ser fluxos obrigatórios, indicando que, ao seguir o caminho de entrada para o nó de bifurcação, todos os caminhos de saída devem ser seguidos. Para que o fluxo principal de execução do processo não seja interrompido, nós de junção são usados para fazer a convergência dos caminhos fluxos paralelos em um único caminho de saída que retoma o fluxo principal. Essa alteração evita dúvidas sobre obrigatoriedade ou não de se seguir os fluxos de saída de um nó de bifurcação, uma vez que a semântica do símbolo determina que é obrigatório seguir todos os caminhos de saída. A Figura 6.7 mostra um exemplo desta alteração.

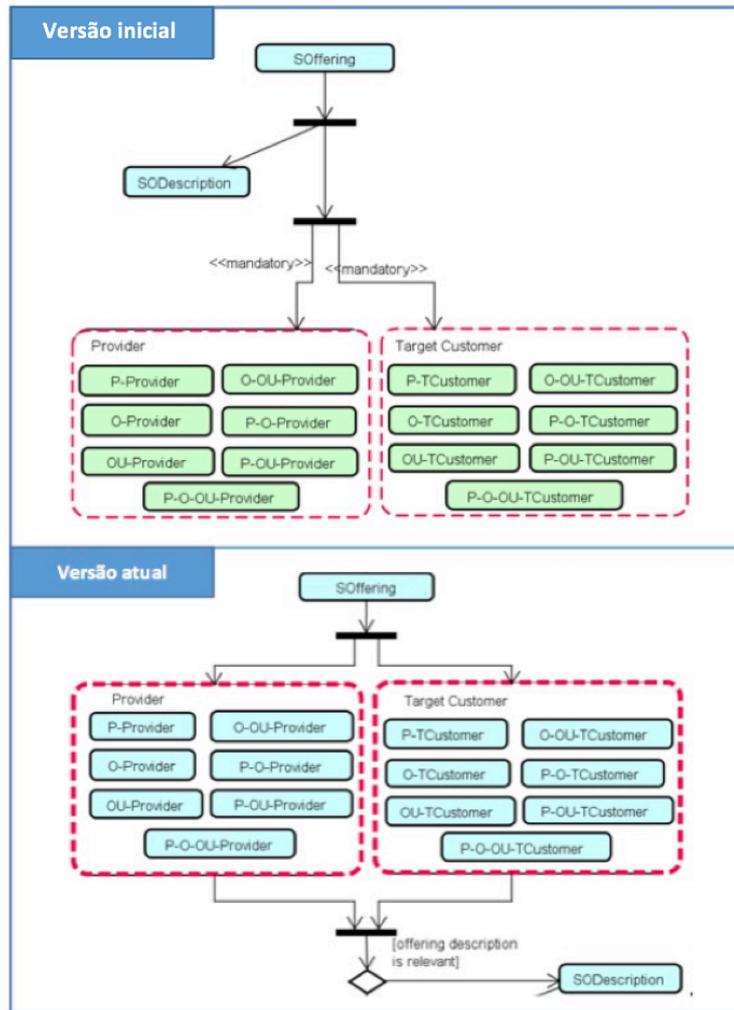


Figura 6.7 - Mudanças no uso do *Fork Node* e adição do *Join Node*.

(iv) Ações de Aplicação

Na versão inicial de S-OPL, o símbolo correspondente a nós de ação (*Action Nodes*) em UML eram usados para representar padrões. Na versão atual de S-OPL, o modelo de processo é composto por ações de aplicação dos padrões e, portanto, o símbolo correspondente a nós de ação em UML (retângulo arredondado com rótulo) representa ações de aplicação de padrões. De forma similar, os grupos de padrões e grupos de padrões variantes passaram a ser grupos de ações de aplicação de padrões e grupos de ações de aplicação de padrões variantes.

(vi) Gerenciamento de Complexidade

Para realizar gerenciamento de complexidade na nova versão de S-OPL, foi acrescentado ao modelo de processo detalhado da LPO um modelo mais geral, onde os grupos foram representados no formato caixa-preta, fornecendo uma visão mais clara do

processo como um todo. Assim, S-OPL passou a ter um diagrama com a gerência de complexidade (visão caixa-preta) e o outro mostrando o conteúdo interno aos grupos (visão expandida). A Figura 6.8 apresenta o Modelo de Processo de S-OPL no formato caixa-preta. Segundo as orientações apresentadas na Seção 6.3, diferentes cores são usadas para identificar os padrões de diferentes grupos. A Figura 6.9 apresenta o Modelo de Processo de S-OPL no formato expandido.

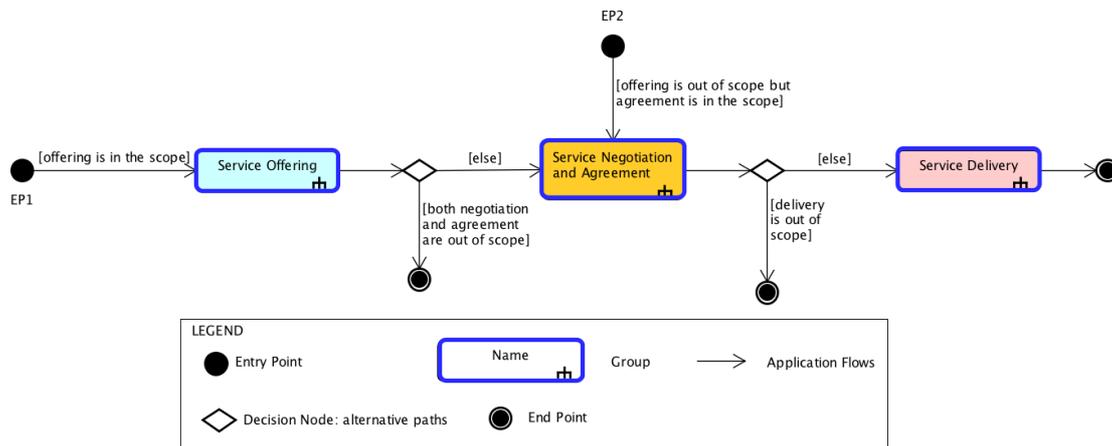


Figura 6.8 - Modelo de Processo de S-OPL – com Gerência de Complexidade (formato caixa-preta).

Durante a reengenharia de S-OPL, além das alterações realizadas devido à utilização de OPL-ML, também foram realizadas alterações no projeto da LPO. Essas alterações são explicadas a seguir.

(i) Alteração da localização dos grupos de padrões variantes

Na versão inicial de S-OPL, o engenheiro de ontologias, após selecionar o padrão *SOffering*, precisava escolher um tipo de prestador de serviço e um tipo de cliente alvo. Para isso, ele seguia um caminho até o grupo Prestador de Serviço e Cliente (*Service Provider and Customer*) e depois precisava retornar ao fluxo principal. Para evitar quebra no sequenciamento do fluxo no processo, o grupo Prestador de Serviço e Cliente (*Service Provider and Customer*) foi eliminado e seus subgrupos de padrões variantes foram incluídos nos grupos no contexto em quais devem ser aplicados. Assim, os grupos de padrões variantes *Provider* e *Target Customer* foram incluídos no grupo *Service Offering*, enquanto *Hired Provider* e *Service Customer* foram incluídos no grupo *Service Negotiation and Agreement*. Além dessa alteração realizada visando à melhoria da sequência do fluxo principal, as alterações citadas nos itens anteriores também contribuíram para isso, resultando em um modelo de processo mais claro, com os caminhos a serem seguidos melhor definidos e com um fluxo sequencial principal que guia do início do processo até os nós de finalização. A Figura 6.10 mostra um exemplo da alteração da localização dos grupos de padrões variantes.

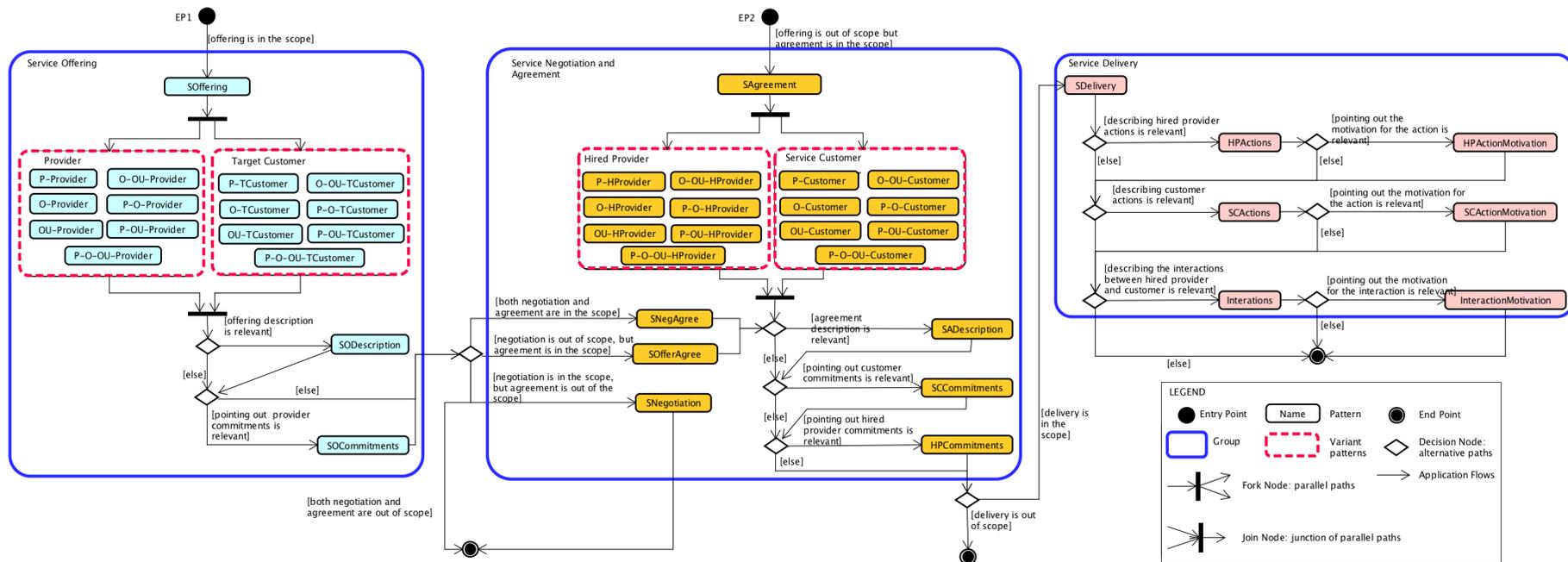


Figura 6.9 - Modelo de Processo de S-OPL (formato expandido)

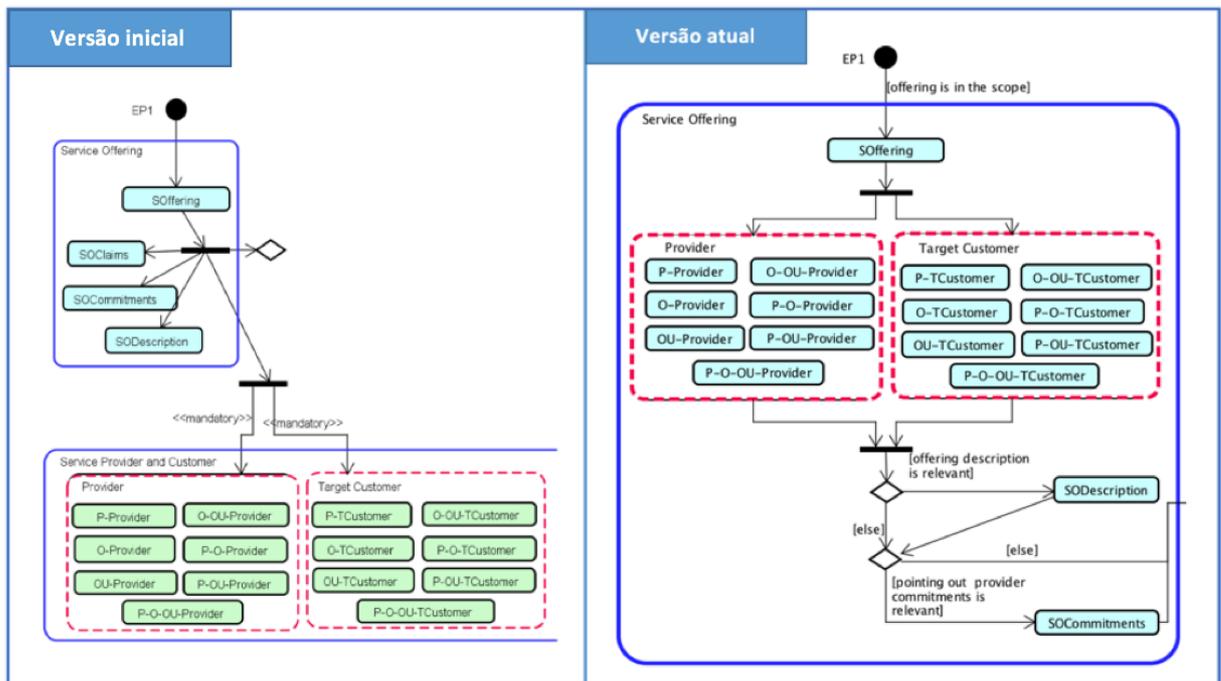


Figura 6.10 - Inclusão dos grupos de padrões variantes Provider e Target Customer no grupo Service Offering.

(ii) Alterações relacionadas ao Domínio

Considerando-se que os padrões que tratam reivindicações (*claims*) são apenas uma visão sob outra perspectiva dos compromissos (*commitments*) (por exemplo, um cliente que aluga o carro tem o compromisso de entregar o carro no prazo acordado e o prestador de serviços tem a reivindicação de que o carro seja entregue no prazo acordado) e, portanto, não tratam problemas diferentes dos tratados por outros padrões de S-OPL, decidiu-se por omiti-los. Dessa forma, os padrões *SOClaims*, *HPClaims* e *SCClaims* foram excluídos.

6.6 Avaliação de OPL-ML

Com o **objetivo** de avaliar OPL-ML, foi conduzido um *survey*. Como informado no capítulo 4, um *survey* é uma pesquisa de opinião que, frequentemente, é realizada em retrospecto. Ela pode ser realizada, por exemplo, após a utilização de uma ferramenta ou técnica (PFLEEGER, 1994). Dessa forma, um *survey* permite capturar um 'retrato instantâneo' de uma situação. Os principais meios de aplicação de um *survey* são questionários e entrevistas, que são realizados com uma amostra representativa da população. Os questionários são respondidos pela amostra selecionada e os resultados são coletados e analisados para derivar uma conclusão. O resultado é, então, generalizado para a população representada (MAFRA; TRAVASSOS, 2006).

O procedimento de realização do *survey* consistiu na aplicação de um questionário contendo questões sobre a nova representação de S-OPL. O questionário foi aplicado no contexto da disciplina Engenharia de Ontologias ministrada no segundo semestre de 2015.

O **questionário** usado contém as mesmas perguntas relacionadas à representação visual presentes no questionário usado anteriormente nos estudos experimentais (vide capítulos 4 e 5), tendo sido acrescentadas quatro perguntas relacionadas às diferenças entre a versão de S-OPL usada no experimento e a nova versão disponibilizada para avaliação no contexto do *survey*. Para elaboração do questionário eletrônico foi utilizado o *Google Forms*². Os formulários utilizados podem ser vistos na íntegra no Apêndice A. A Figura 6.11 apresenta um fragmento do questionário utilizado, incluindo algumas das questões relacionadas à comparação da versão atual de S-OPL com a inicial.

Núcleo de Estudos em Modelagem
Conceitual e Ontologias
Departamento de Informática, Universidade Federal do Espírito Santo, Brasil

Estudo Experimental

* Required

5) Você considera que essa versão tornou o processo mais claro? *

Muito menos claro

Menos claro

Mantém a mesma clareza

Mais claro

Bem mais claro

*
Justifique sua resposta

6) Em relação à facilidade de entendimento do processo da OPL, qual representação você julga melhor? *

A antiga, que permitia apenas mostrar o processo como um todo;

A nova, que contempla diferentes níveis de abstração, permitindo trabalhar com uma visão geral do processo e com processos detalhados, que podem ser trabalhados no todo ou por grupos.

*
Justifique sua resposta

Figura 6.11 - Fragmento do Questionário de Avaliação usado no *Survey*.

Os **participantes** do *survey* foram 6 alunos dos 9 participantes do segundo estudo experimental que avaliou a versão inicial de S-OPL (vide Capítulo 5). Dentre os participantes, 5 declararam ter experiência *média* com modelagem conceitual. O outro participante declarou ter *baixa* experiência. Em relação à experiência em desenvolvimento de ontologias, 2 participantes declararam ter *média* experiência, 2 declararam ter experiência *baixa* e 2

² <http://www.google.com/forms/about/>

declararam não ter experiência. Por fim, 4 participantes responderam ter *baixa* experiência com LPOs e 2 declararam não possuir experiência.

Novamente decidiu-se não excluir os participantes com pouca experiência com LPOs ou com desenvolvimento de ontologias, uma vez que todos possuem um conhecimento mínimo em modelagem conceitual e refletem os possíveis perfis de usuários de LPOs.

6.6.1 Resultados do Survey

Nesta seção são apresentados os principais resultados obtidos a partir dos questionários respondidos pelos participantes.

O questionário inicia-se com perguntas relacionadas à LPO propriamente dita, visando capturar a percepção dos participantes quanto ao entendimento do funcionamento da LPO e dos símbolos utilizados na sua notação visual.

6.6.1.1 Percepção acerca do entendimento do mecanismo de funcionamento geral de LPOs.

Embora os participantes do estudo tenham *baixa* ou tiveram sua *primeira* experiência com LPOs nesse estudo, a grande maioria considerou o entendimento do novo mecanismo de funcionamento da LPO *fácil* (4 participantes) ou *muito fácil* (1 participante). Apenas 1 considerou *neutro*. Esse participante declarou ter *média* experiência com modelagem conceitual e ontologias e *baixa* experiência com LPOs. Dentre os alunos que consideraram o mecanismo de funcionamento da LPO *fácil*, um declarou ter *baixa* experiência com modelagem conceitual e os outros cinco participantes declararam ter experiência *média*. Em relação a ontologias, 2 participantes informaram ter *nenhuma* experiência, tendo apenas 2 com experiência *média* e 2 com experiência *baixa*. Quanto à experiência com LPOs, 4 responderam ter *baixa* experiência e os outros 2 declararam ter *nenhuma* experiência. Visto isto, pode-se afirmar que não é necessário ter muita experiência com ontologias ou LPOs para entender o mecanismo de LPOs.

Com relação aos símbolos utilizados para representar os elementos da LPO, houve uma boa aceitação por parte da maioria dos participantes. Um aluno considerou o símbolo usado para representar Grupo de Padrões Variantes de *difícil* compreensão e o símbolo Nó de Decisão *neutro*. Esse participante declarou ter *média* experiência em modelagem conceitual e ontologias, e *baixa* experiência com LPOs. A Figura 6.12 apresenta as respostas sobre esse tópico.

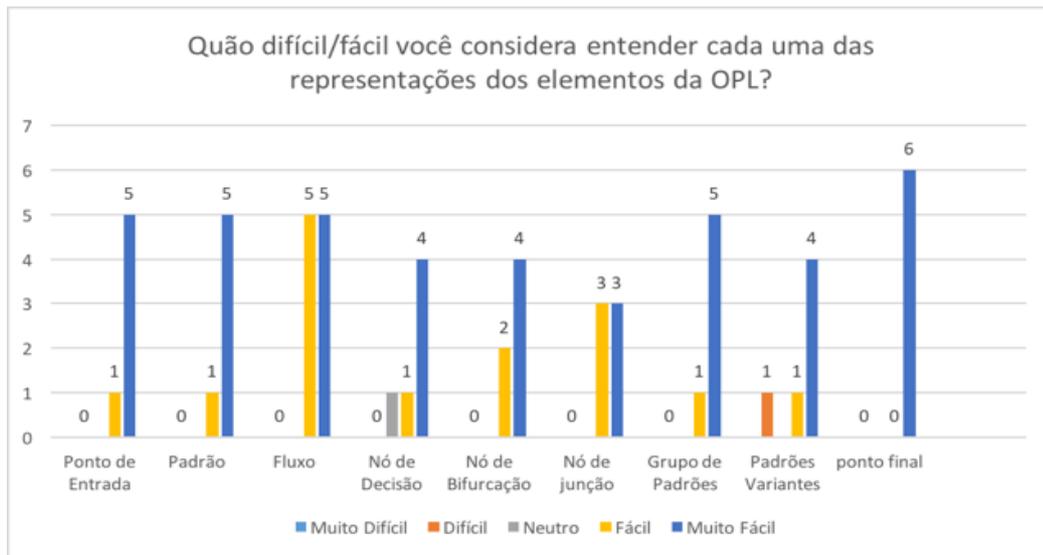


Figura 6.12 Compreensão dos símbolos da nova representação de S-OPL.

Um participante indicou ter tido alguma dificuldade para entender o símbolo usado para representar Grupo de Padrões Variantes e o Nó de Decisão. O restante dos participantes declarou que não teve dificuldades em entender os símbolos.

Nenhuma sugestão de melhoria foi apresentada e os participantes destacaram que na nova versão de S-OPL o processo e as relações entre os padrões estão bem explicitados e os problemas identificados na versão anterior foram resolvidos.

6.6.1.2 *Percepção acerca das Mudanças realizadas na LPO*

A segunda parte do questionário visou investigar se as mudanças realizadas na representação da LPOs foram satisfatórias.

Todos os participantes declararam que a nova versão de S-OPL tornou o processo de aplicação dos padrões muito mais claro (5 participantes) ou mais claro (1 participante) do que a versão anterior. Eles destacaram que a nova versão permite continuidade do fluxo principal e maximiza o entendimento do modelo. A Figura 6.13 apresenta esses resultados.

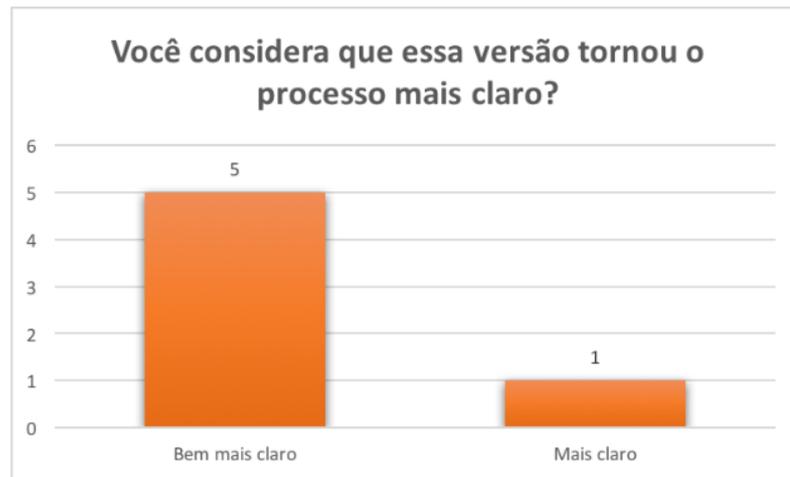


Figura 6.13 – Análise do impacto da nova representação no processo.

Na questão “Em relação à facilidade de entendimento do processo da LPO, qual representação você julga melhor?”, todos os participantes responderam que a nova versão, que contempla diferentes níveis de abstração, permitindo trabalhar com uma visão geral do processo e com processos detalhados, que podem ser trabalhados no todo ou por grupos, permite entender o processo da LPO mais facilmente.

Na questão “Em relação ao modelo que apresenta os componentes da S-OPL e as relações de dependência entre eles, você considera que ele facilita ou dificulta a aplicação e o entendimento da S-OPL?”, que se refere ao modelo estrutural, 5 participantes declararam que o modelo facilita e 1 declarou que *facilita muito*. Para eles, esse novo recurso facilitou ainda mais o entendimento do processo e serve como um mapa para guiar os usuários nas relações existentes entre os padrões, minimizando, assim, a necessidade de consultas na documentação textual da LPO.

6.6.2 Discussões

O foco do *survey* foi verificar se as alterações realizadas na notação visual utilizada surtiram efeito positivo, ou seja, se utilizando-se OPL-ML, o entendimento acerca de S-OPL foi melhorado. Para isso, deve-se avaliar se houve mudança no grau de dificuldade para entendimento do mecanismo de funcionamento da LPO, no grau de dificuldade para entendimento dos símbolos da LPO e se há algum problema na representação da notação.

Os resultados do estudo indicam um baixo grau de dificuldade para entendimento do mecanismo de funcionamento da LPO, bem como para entender a maioria dos símbolos da notação visual. Além disso, não foram relatados problemas na nova representação da LPO e a nova representação foi considerada melhor que a anterior.

Os resultados obtidos podem ser entendidos como indícios de que OPL-ML é adequada para a representação de LPOs. Embora os resultados sejam preliminares, pode-se considerar que eles corroboram a afirmação de que OPL-ML é de fácil entendimento e aprimora a utilização de LPOs no desenvolvimento de ontologias de domínio.

6.6.3 Ameaças à Validade

O *survey* foi realizado em um contexto muito similar ao dos experimentos descritos nos capítulos 4 e 5, ou seja, os participantes foram alunos do Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo e a aplicação do questionário foi realizada no contexto de uma disciplina. Dessa forma, as ameaças envolvidas neste *survey* são praticamente as mesmas presentes nos estudos experimentais realizados, sendo acrescida a seguinte ameaça: o fato de os participantes terem tido contato com uma versão inicial de S-OPL antes de avaliar a segunda versão pode influenciar na avaliação da segunda versão da LPO, uma vez que algum conhecimento sobre a LPO já foi adquirido, podendo levar os participantes a identificarem a segunda versão como melhor não apenas pelas melhorias que ela apresenta, mas também por terem conhecimento prévio sobre o que é tratado na LPO.

As ameaças presentes no estudo limitam a possibilidade de generalização dos resultados. Por isso, os resultados do estudo não podem ser generalizados e são considerados apenas resultados preliminares e indícios, não sendo conclusivos.

6.7 Considerações finais

Este capítulo apresentou a principal contribuição deste trabalho: OPL-ML, uma notação visual para representar Linguagens de Padrões Ontológicos. A notação proposta considera LPOs sob duas perspectivas, estrutural e comportamental, e permite a representação de dois tipos de modelo: modelo de processo e modelo estrutural. A notação visual é composta por uma sintaxe abstrata, que define os constructos semânticos da notação, e uma sintaxe concreta, que define os símbolos para representar visualmente os constructos.

A sintaxe abstrata baseia-se em um fragmento do metamodelo do diagrama de atividades da UML. No desenvolvimento da sintaxe concreta enfatizou-se o fato de as linguagens de padrões necessitarem ser representadas de forma cognitivamente rica. Visto que os princípios de PoN auxiliam no *design*, criação e reestruturação de notações visuais de forma a aprimorar sua efetividade cognitiva, decidiu-se projetar a sintaxe concreta das notações visuais dos modelos de LPOs valendo-se desses princípios. Esses princípios foram

considerados aplicando-se um processo de sistematização de PoN definido em (TEIXEIRA *et al.*, 2016).

Resultados do mapeamento sistemático (Capítulo 3) e dos experimentos (capítulos 4 e 5) também foram úteis no desenvolvimento da sintaxe concreta. Alguns símbolos da notação concreta (por exemplo, os símbolos adotados para representar padrão e fluxo) são consistentes com os resultados encontrados no mapeamento sistemático. Dentre os resultados do experimento que contribuiriam para a definição da sintaxe concreta pode-se destacar o símbolo usado para representar padrões variantes, que resultou de uma alteração no símbolo usado na LPO usada no experimento, realizada a partir de comentários dos participantes.

Como prova de conceito, OPL-ML foi utilizada na reengenharia de uma linguagem de padrões ontológicos de serviços (S-OPL – *Service Ontology Pattern Language*) (FALBO *et al.*, 2016). Como avaliação complementar, foi realizado um *survey* considerando a nova versão de S-OPL, para analisar a percepção da notação usada e melhorias realizadas sob a perspectiva de usuários de OPL-ML. Os resultados do *survey* podem ser entendidos como indícios de que OPL-ML é viável e capaz de representar adequadamente LPOs. No entanto, vale ressaltar que os resultados foram obtidos em um *survey* que apresenta algumas limitações. Algumas foram tratadas como ameaças à validade, mas algumas não são possíveis de serem tratadas (por exemplo, o fato de o estudo ter sido realizado em ambiente acadêmico). Sendo assim, novas avaliações de OPL-ML devem ser realizadas para avaliar sua adequação e aprimorá-la.

Capítulo 7

Conclusão

Neste capítulo são feitas as considerações finais deste trabalho (Seção 7.1), sendo apresentadas suas principais contribuições (Seção 7.2) e perspectivas de trabalhos futuros para continuidade e aprimoramento da pesquisa (Seção 7.3).

7.1 Considerações Finais

Uma Linguagem de Padrões Ontológicos (LPO) é uma rede de padrões ontológicos interconectados, capaz de fornecer suporte holístico para o desenvolvimento de ontologias em um dado campo (FALBO *et al.*, 2016). O uso de LPOs na Engenharia de Ontologias permite melhorar a produtividade do processo de desenvolvimento, diminuindo tempo e custos, e contribui para melhorar a qualidade das ontologias produzidas (RUY *et al.*, 2015b).

Uma LPO organiza padrões ontológicos em uma rede de forma que possam ser usados como blocos de construção que podem ser combinados para tratar os problemas relevantes no desenvolvimento de uma ontologia para o domínio da LPO. Além da rede de padrões, uma LPO provê um processo que guia a ordem de aplicação dos padrões e sugere quais padrões utilizar de acordo com os problemas a serem tratados. O uso de LPOs na Engenharia de Ontologias pode auxiliar engenheiros de ontologias a lidarem com a complexidade envolvida nessa atividade (FALBO *et al.*, 2016).

Pela sua natureza, em linguagens de padrões, não basta apresentar um conjunto ilimitado de soluções sustentáveis para um assunto específico: é necessário iniciar um diálogo com seus usuários para orientá-los através dos espaços de soluções que a abrangem. Este diálogo apresenta os desafios das linguagens sendo abordados em seus espaços de solução, motiva potenciais projetos e alternativas para a resolução de problemas específicos, e incentiva decisões explícitas na presença de tais alternativas (DEUTSCH, 2004).

O diálogo iniciado por linguagens de padrões produz um efeito mais abrangente e mais profundo do que aqueles iniciados por padrões independentes ou organizados em catálogos. Por exemplo, padrões independentes apresentam e discutem o conhecimento sobre como resolver um problema específico, local. Embora este breve diálogo seja melhor do que nenhuma discussão, ele tem efeito e aplicabilidades limitados.

Segundo Moody (2009), notações visuais desempenham papel importante na comunicação, permitindo a representação de abstrações que contribuem para a visualização

de informações que podem levar a um entendimento mais eficaz do que o obtido a partir da leitura de textos. Nesse sentido, LPOs devem ser representadas visualmente por meio de notações visuais que permitam a representação dos padrões propriamente ditos, das relações entre eles e do processo que guia sua aplicação. Em outras palavras, a representação visual de uma LPO deve ser cognitivamente rica, permitindo fácil entendimento dos elementos presentes na LPO, bem como das relações entre eles e da ordem de aplicação de padrões.

Considerando que o uso de LPOs é uma abordagem recente e promissora na Engenharia de Ontologias e que não há uma forma de representação visual estabelecida para elas, neste trabalho foi explorado o tema notações visuais para LPOs.

Buscando-se conhecimento na Engenharia de Software, área onde linguagens de padrões (LPs) têm sido estudadas há anos, foi realizada uma investigação das notações visuais utilizadas para representar LPs de Software. Os resultados do mapeamento sistemático conduzido permitiram a identificação de algumas lacunas, entre elas: (i) não há padrão para as notações visuais das LPs; (ii) as notações visuais adotadas são, em geral, cognitivamente pobres; (iii) não há preocupação em prover mecanismos para auxiliar a seleção dos padrões; e (iv) ausência de uma visão completa das LPs, que apresente aspectos relacionados ao processo da LP e à sua estrutura.

Ainda no contexto deste trabalho, a notação utilizada até então para a representação visual de LPOs foi avaliada através de dois estudos experimentais. Essa análise apontou algumas lacunas, tais como: (i) a falta de clareza de alguns símbolos da representação e (ii) a dificuldade para entendimento de alguns fluxos do processo.

Considerando essas lacunas, este trabalho teve como objetivo definir uma notação para representação visual de linguagens de padrões ontológicos. Para isso, foram utilizados os resultados obtidos no mapeamento sistemático, os resultados obtidos nos estudos experimentais e PoN-S. *PoN Systematized* (PoN-S) (TEIXEIRA *et al.*, 2016) é uma abordagem sistemática que une a teoria e prática da PoN (*Physics of Notation*) (MOODY, 2009) no projeto de sintaxes concretas para linguagens visuais de modelagem. PoN-S auxilia no projeto de notações visuais cognitivamente efetivas através da definição de um processo de *design* que estabelece atividades a serem realizadas, a ligação delas aos princípios PoN, bem como critérios para agrupar os princípios PoN que orienta este processo.

O objetivo geral deste trabalho foi detalhado em quatro objetivos específicos, sendo que todos foram alcançados neste trabalho. A Tabela 7.1 apresenta os objetivos específicos do trabalho e o principal produto que serve como evidência do alcance de cada objetivo.

Tabela 7.1 - Objetivos específicos do trabalho.

Objetivos	Produto
Investigar o estado da arte acerca de representação visual de linguagens de padrões no âmbito de Software;	Mapeamento Sistemático da Literatura (vide Capítulo 3)
Desenvolver uma Linguagem de Padrões Ontológicos utilizando a notação visual usada até então em trabalhos do NEMO e identificar problemas dessa notação;	S-OPL (vide Capítulo 5)
Definir uma sintaxe abstrata que identifique os constructos semânticos para tratar aspectos estrutural e comportamental de linguagens de padrões ontológicos;	Sintaxe abstrata de OPL-ML. (vide Capítulo 6)
Definir uma sintaxe concreta para representação de aspectos estrutural e comportamental de linguagens de padrões ontológicos;	Sintaxe concreta de OPL-ML. (vide Capítulo 6)
Utilizar a notação proposta para representar uma Linguagem de Padrões Ontológicos desenvolvida no NEMO e analisar as melhorias obtidas em relação à notação utilizada anteriormente.	Reengenharia de S-OPL e <i>Survey</i> . (vide Capítulo 6)

Ao se comparar a notação visual de LPOs proposta neste trabalho e a versão inicial da notação visual utilizada nos trabalhos publicados no NEMO, percebe-se que grandes avanços foram alcançados.

Primeiramente, a notação proposta definiu claramente as sintaxes abstrata e concreta, formalizando a linguagem visual. A sintaxe abstrata foi definida de forma bem fundamentada, analisando os elementos essenciais para representação de linguagens de padrões de software (baseando-se nos resultados do mapeamento sistemático) e definindo um metamodelo, parcialmente baseado num metamodelo de uma linguagem amplamente usada na modelagem conceitual de projetos na Engenharia de Software, o diagrama de atividades da UML.

A sintaxe concreta foi desenvolvida através de um processo sistematizado que aplicou os princípios de PoN, de forma a garantir uma notação visual cognitivamente rica. A definição de um modelo para as relações entre padrões separado do modelo do processo de aplicação dos padrões facilitou ainda mais o entendimento do mecanismo de LPOs, possibilitando, assim, uma aplicação mais prática dos padrões ontológicos no desenvolvimento de ontologias de domínio.

Outro recurso adicionado à linguagem visual, que aprimorou o processo de entendimento das LPOs, foi o gerenciamento da complexidade dos modelos através de diagramas com dois níveis de abstração: um com a visão geral do processo (apresentação dos

grupos de padrões em formato “caixa-preta”) e outro com uma visão detalhada do processo (apresentação dos grupos de padrões em formato expandido).

A prova de conceito realizada com S-OPL mostrou esses avanços na linguagem visual. Além disso, o *survey* apresentado no Capítulo 6 capturou a percepção dos participantes quanto a esses novos recursos da notação visual de LPOs. Nas declarações dos participantes, destacou-se as impressões positivas relativas à nova notação visual utilizada nos modelos de LPOs e na aplicação de LPOs no desenvolvimento de ontologias. Quanto à notação visual, destacaram ser uma notação clara e de fácil entendimento. Em relação ao processo de desenvolvimento com LPOs, enfatizaram ser intuitivo, prático e produtivo.

Entre as limitações deste trabalho, pode ser destacada a avaliação de OPL-ML, que foi realizada com poucos participantes e em ambiente acadêmico. Dessa forma, os resultados da avaliação não podem ser considerados conclusivos, mas apenas indícios de que a notação visual é viável e eficiente no apoio ao desenvolvimento de ontologias de domínio.

Vale ressaltar que a definição de uma notação visual para LPOs amplia os recursos disponíveis para os engenheiros de ontologias no desenvolvimento de ontologias de domínio, aumentando a produtividade e possibilitando maiores avanços na área de LPOs.

7.2 Contribuições

As principais contribuições desta dissertação são:

- (i) Mapeamento Sistemático da Literatura, que descreve o panorama das notações visuais das Linguagens de Padrões de Software.
- (ii) S-OPL, uma linguagem de padrões ontológicos que pode ser utilizada como base para o desenvolvimento de ontologias de serviços em domínios específicos.
- (iii) A Notação Visual de Linguagens de Padrões Ontológicos, composta por um Modelo de Processo de Aplicação dos Padrões e um Modelo Estrutural.

7.3 Trabalhos Futuros

Considerando a pesquisa aqui apresentada, há algumas perspectivas de trabalhos futuros. No âmbito da *pesquisa* pode-se destacar:

- (i) Atualizar e aprofundar a investigação da literatura realizada, através da aplicação de *snowballing*, a fim de verificar o surgimento de novas Linguagens de Padrões de Software.

- (ii) Realizar a reengenharia de outras LPOs existentes: SP-OPL (FALBO *et al.*, 2013), ISP-OPL (RUY *et al.*, 2015a), E-OPL (FALBO *et al.*, 2014) e M-OPL (BARCELLOS *et al.*, 2014).
- (iii) Definir uma abordagem para o desenvolvimento de LPOs usando a notação proposta, de modo a explicitar um conjunto de boas práticas para o desenvolvimento de OPLs.
- (iv) Definir uma abordagem para a aplicação de LPOs no desenvolvimento de ontologias de domínio.
- (v) Investigar as relações entre padrões presentes nas LPs analisadas no mapeamento sistemático e que não foram identificadas no contexto de LPOs.

Em relação à *avaliação da notação visual de LPOs*, pode-se:

- (i) Realizar novos estudos para avaliar a proposta, com uma maior quantidade de participantes e maior variedade de perfis.
- (ii) Realizar estudos de caso do uso de LPOs no desenvolvimento de ontologias de domínio, com a notação proposta, em um contexto real em uma organização. Embora S-OPL tenha sido utilizada em um caso real (FALBO *et al.*, 2016), a versão utilizada foi a inicial, que não era representada utilizando-se OPL-ML.

Em relação ao *apoio ao desenvolvimento e aplicação de LPOs*, pode-se:

- (i) Desenvolver uma ferramenta de apoio à criação de LPOs, que auxilie na criação dos modelos das LPOs, e à utilização de LPOs, que auxilie na integração dos padrões, em nível de instância dos modelos, durante o processo de aplicação dos padrões de uma LPO.

Referências Bibliográficas

- AGUIAR, A.; DAVID, G. **Patterns for documenting frameworks - Part II**. EuroPLOP. p.393–406, 2006.
- AHLUWALIA, K. S. **Scalability design patterns**. 14th Conference on Pattern Languages of Programs. 2007. ACM.
- AMATRIAIN, X.; ARUMI, P. **Frameworks generate domain-specific languages: A case study in the multimedia domain**. Software Engineering, IEEE Transactions on, v. 37, n. 4, p. 544–558, 2011.
- AVGERIOU, P.; TANDLER, P. **Architectural patterns for collaborative applications**. International journal of computer applications in technology, v. 25, n. 2-3, p. 86–101, 2006.
- AVGERIOU, P.; ZDUN, U. **Architectural patterns revisited : A pattern language**. EuroPLOP. p.431–470, 2005. UVK Konstanz.
- BARCELLOS, M. P.; FALBO, R. DE A.; FRAUCHES, V. G. **Towards on a Measurement Ontology Pattern Language**. First Joint Workshop Onto.Com/ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering. 2014. Rio de Janeiro - RJ, Brazil.
- BARROS, M. DE O.; WERNER, C. M. L.; TRAVASSOS, G. H. **Um Estudo Experimental sobre a Utilização de Modelagem e Simulação no Apoio à Gerência de Projetos de Software**. XVI Simpósio Brasileiro de Engenharia de Software, p. 191–206, 2002.
- BASIL, V. R.; CALDIERA, G.; ROMBACH, H. D. **Goal Question Metric Approach**. Encyclopedia of Software Engineering, 1994. Hoboken, NJ, USA: John Wiley & Sons, Inc.
- BELLEBIA, D.; DOUIN, J. M. **Applying patterns to build a lightweight middleware for embedded systems**. Conference on Pattern languages of programs. 2006. ACM.
- BOONE, S.; BERNAERT, M.; ROELENS, B.; MERTENS, S.; POELS, G. **Evaluating and improving the visualisation of CHOOSE, an enterprise architecture approach for SMEs**. In The practice of enterprise modeling. p.87–102, 2014. Springer Berlin Heidelberg.

- BORCHERS, J. O. **A pattern approach to interaction design**. *Ai & Society*, v. 15, n. 4, p. 59–376, 2001.
- BRAGA, M. R.; BEZERRA, C. I.; MONTEIRO, J. M. S.; ANDRADE, R. **A pattern language for agile software estimation**. 9th Latin-American Conference on Pattern Languages of Programming. 2012.
- BRAGA, R. T.; MASIERO, P. C. **Finding frameworks hot spots in pattern languages**. *Journal of Object Technology*, v. 3, n. 1, p. 123–142, 2004.
- BRICKLEY, D.; MILLER, L. **FOAF Vocabulary Specification**. 2005.
- BUSCHMANN, F. **Applying patterns**. Disponível em: <http://www.cs.wustl.edu/~schmidt/PDF/applying-patterns.pdf>. .
- BUSCHMANN, F.; HENNEY, K.; SCHIMDT, D. **Pattern-oriented Software Architecture: On Patterns and Pattern Language**. John Wiley & Sons Ltd., 2007.
- DELESSY, N.; FERNANDEZ, E. B.; LARRONDO-PETRIE, M. M. **A Pattern Language for Identity Management**. *Computing in the Global Information Technology, 2007. ICCGI 2007. International Multi-Conference on*. p.31, 2007. IEEE.
- DEUTSCH, P. **Models and Patterns**. In: J. Greenfield; K. Short; S. Cook; S. Kent (Orgs.); *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*, 2004. Indianapolis: John Wiley & Sons.
- DING, Y. **A review of ontologies with the Semantic Web in view**. *Journal of Information Science*, v. 27, p. 377 – 384, 2001.
- ELORANTA, V. P.; KOSKINEN, J.; LEPPÄNEN, M.; REIJONEN, V. **Software architecture patterns for distributed machine control systems**. EuroPLOP. 2009.
- ELORANTA, V.-P.; LEPPÄNEN, M. **Human machine interface patterns for distributed machine control systems**. 17th European Conference on Pattern Languages of Programs. p.3, 2012. ACM.
- FALBO, R. DE A.; BARCELLOS, M. P.; NARDI, J. C.; GUIZZARDI, G. **Organizing ontology design patterns as ontology pattern language**. *Proceedings of the 10th Extended Semantic Web Conference - ESWC 2013-a*. 2013. Montpellier, France.
- FALBO, R. DE A.; QUIRINO, G. K.; NARDI, J.; et al. **An Ontology Pattern Language for Service Modeling**. *The 31st ACM/SIGAPP Symposium on Applied Computing*.

2016. Pisa, Italy.

FALBO, R. DE A.; RUY, F. B.; GUIZZARDI, G.; BARCELLOS, M. P.; ALMEIDA, J. P. A. **Towards an enterprise ontology pattern language**. Proceedings of the 29th Annual ACM Symposium on Applied Computing. p.323–330, 2014. ACM.

FALBO, R. DE A.; GUIZZARDI, G.; GANGEMI, A.; PRESUTTI, V. **Ontology Patterns: Clarifying Concepts and Terminology**. Proceedings of the 4th Workshop on Ontology and Semantic Web Patterns. 2013. Sydney, Australia.

FOGLI, D.; PROVENZA, L. P.; BERNAREGGI, C. **A universal design resource for rich Internet applications based on design patterns**. Universal access in the information society, v. 13, n. 2, p. 205–226, 2014.

FOGLI, D.; PROVENZA, L. P.; COLOSIO, S. **Metadesigning e-government services: a case study in a local agency**. AVI. 2010.

GANGEMI, A.; PRESUTTI, V. **Ontology Design Patterns**. In: S. Staab; R. Studer (Orgs.); Handbook on Ontologies. Second ed., p.221 – 243, 2009. Springer.

GOEDICKE, M.; KÖLLMANN, C.; ZDUN, U. **Designing runtime variation points in product line architectures: Three cases**. Science of Computer Programming, v. 53, n. 3, p. 353–380, 2004.

GOEDICKE, M.; ZDUN, U. **Piecemeal legacy migrating with an architectural pattern language: A case study**. Journal of Software Maintenance and Evolution: Research and Practice, v. 14, n. 1, p. 1–30, 2002.

GROLIMUND, D.; MÜLLER, P. **A Pattern Language for Overlay Networks in Peer-to-Peer Systems**. EuroPLOP. p.95–140, 2006.

GRUBER, T. R. **A translation approach to portable ontology specifications**. Knowledge Acquisition, v. 5, n. 2, p. 199–220, 1993.

GUARINO, N. **Formal Ontology and Information Systems., In Formal Ontology in Information Systems**. In: N. (ed. . Guarino (Org.); Proceedings of the International Conference on Formal Ontology and Information Systems (FOIS). p.3–15, 1998. Trento, Italy: June 6-8, IOS Press, Amsterda.

GUERRA, E.; ALVES, F.; KULESZA, U.; FERNANDES, C. **A reference architecture for organizing the internal structure of metadata-based frameworks**. Journal of

Systems and Software, v. 86, n. 5, p. 1239–1256, 2013., 2013.

GUIZZARDI, G. **Ontological Foundations for Structural Conceptual Models**, Universal Press., 2005.

GUIZZARDI, G. **On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models**. Frontiers in Artificial Intelligence and Applications, Databases and Information Systems IV. IOS Press, Amsterdam., v. 155, p. 18–39, 2007.

GURR, C. A. **Effective diagrammatic communication: Syntactic, semantic and pragmatic issues**. Journal of Visual Languages & Computing, v. 10, n. 4, p. 317–342, 1999.

GZARA, L.; RIEU, D.; TOLLENAERE, M. **Product information systems engineering: An approach for building product models by reuse of patterns**. Robotics and Computer-Integrated Manufacturing, v. 19, n. 3, p. 239–261, 2003.

HAFIZ, M.; ADAMCZYK, P.; JOHNSON, R. E. **Growing a pattern language (for security)**. ACM international symposium on New ideas, new paradigms, and reflections on programming and software. p.139–158, 2012.

HANMER, R. S.; KOCAN, K. F. **Documenting architectures with patterns**. Bell Labs Technical Journal, v. 9, n. 1, p. 143–163, 2004.

HANNEBAUER, C.; WOLFF-MARTING, V.; GRUHN, V. **Towards a pattern language for FLOSS development**. 17th Conference on Pattern Languages of Programs. 2010.

HENTRICH, C. **Synchronization patterns for process-driven and service-oriented architectures**. Transactions on Pattern Languages of Programming I, p. 103–135, 2009.

HENTRICH, C.; ZDUN, U. **Service Integration Patterns for Invoking Services from Business Processes**. EuroPLOP. p.235–278, 2007.

HENTRICH, C.; ZDUN, U.; HLUPIC, V.; DOTSIKA, F. **An Approach for Pattern Mining Through Grounded Theory Techniques and Its Applications to Process-driven SOA Patterns**. 18th European Conference on Pattern Languages of Program. p.9, 2015. ACM.

HÖLZL, M.; KOCH, N.; MAYER, P.; WIRSING, M. **Sensoria Patterns**. Rigorous

software engineering for service-oriented systems. p.719–736, 2011. Springer Berlin Heidelberg.

HSIEH, C.-Y.; CHENG, Y. C.; CHEN, C.-T. **Emerging patterns of continuous integration for cross-platform software development**. Proceedings of the 2nd Asian Conference on Pattern Languages of Programs. 2011.

JOHN, B. E.; BASS, L.; GOLDEN, E.; STOLL, P. **A responsibility-based pattern language for usability-supporting architectural patterns**. 1st ACM SIGCHI symposium on Engineering interactive computing systems. 2009.

KHALED, O. M.; HOSNY, H. M. **A pattern-driven software performance model**. 10th IASTED International Conference on Software Engineering and Applications, SEA. p.13–15, 2006.

KIRK, D.; ROPER, M.; WOOD, M. **Identifying and addressing problems in object-oriented framework reuse**. Empirical Software Engineering, v. 12, n. 3, p. 243–274, 2007.

KITCHENHAM, B. A.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering**. 2007.

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

LUKOSCH, S.; SCHÜMMER, T. **Groupware development support with technology patterns**. International Journal of Human-Computer Studies, v. 64, n. 7, p. 599–610, 2006.

LYTRA, I.; SOBERNIG, S.; ZDUN, U. **Architectural decision making for service-based platform integration: A qualitative multi-method study**. Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference on. IEEE. p.111–120, 2012.

MAHEMOFF, M.; HUSSEY, A.; JOHNSTON, L. **Pattern-based reuse of successful designs: usability of safety-critical systems**. Software Engineering Conference. p.31–39, 2001. Australian: IEEE.

MEISTER, J.; REUSSNER, R.; ROHDE, M. **Managing product line variability by patterns**. Object-Oriented and Internet-Based Technologies. p.153–168, 2004.

Springer Berlin Heidelberg.

MENG, X. X.; WANG, Y. S.; SHI, L.; WANG, F. J. **A process pattern language for agile methods**. Software Engineering Conference. APSEC 2007. 14th Asia-Pacific. IEEE. p.374–381, 2007.

MILES, A.; BRICKLEY, D. **SKOS Core Vocabulary Specification**. 2005.

MONTEIRO, M. P.; AGUIAR, A. **Patterns for refactoring to aspects: an incipient pattern language**. Conference on Pattern Languages of Programs. 2007. ACM.

MONTEIRO, P.; MONTEIRO, M. P.; PINGALI, K. **Parallelizing irregular algorithms: a pattern language**. 18th Conference on Pattern Languages of Programs. 2013.

MOODY, D. L. **The “Physics” of Notations: Towards a Scientific Basis for Constructing Visual Notations in Software Engineering**. IEEE Transactions on Software Engineering, v. 35, n. 5, p. 756–778, 2009.

MOODY, D. L.; HEYMANS, P.; MATULEVIČIUS, R. **Visual syntax does matter: improving the cognitive effectiveness of the i* visual notation**. Requirements Engineering, v. 15, n. 2, p. 141–175, 2010.

MOURATIDIS, H.; WEISS, M.; GIORGINI, P. **Security patterns meet agent oriented software engineering: a complementary solution for developing secure information systems**. Conceptual Modeling–ER. p.225–240, 2005. Springer Berlin Heidelberg.

MUIJNCK-HUGHES, J. DE; DUNCAN, I. **Thinking Towards a Pattern Language for Predicate Based Encryption Crypto-Systems**. Software Security and Reliability Companion (SERE-C), 2012 IEEE Sixth International Conference on. p.27–32, 2012.

NARDI, J. C.; FALBO, R. A.; ALMEIDA, J. P. A.; GUIZZARDI, G.; PIRES, L. F.; VAN SINDEREN, M. J.; GUARINO, N.; FONSECA, C. M. **A Commitment-based Reference Ontology for Services**. Information Systems, v. 54, p. 263–288, 2015.

NARDI, J. C.; FALBO, R. DE A.; ALMEIDA, J. P. A.; GUIZZARDI, G.; PIRES, L. F.; SINDEREN, M. VAN; GUARINO, N. **Towards a Commitment-based Reference Ontology for Services**. 17th IEEE International Enterprise Distributed Object Computing Conference (EDOC). p.175–184, 2013.

NOBLE, J. **Towards a pattern language for object oriented design**. Technology of

- Object-Oriented Languages, 1998. TOOLS 28. Proceedings. p.2–13, 1998.
- OATES, B. J. **Researching Information Systems and Computing**. SAGE Publications, 2006.
- OMG. **Normative Document of UML 2.5**. Disponível em: <<http://www.omg.org/spec/UML/2.5/PDF>>. .
- PFLEEGER, S. L. **Design and Analysis in Software Engineering**. Acm Sigsoft, v. 19, n. 4, p. 16–20, 1994.
- POVEDA-VILLALÓN, M.; SUÁREZ-FIGUEROA, M. C.; GÓMEZ-PÉREZ, A. **Reusing Ontology Design Patterns in a Context Ontology Network**. Second Workshop on Ontology Patterns (WOP 2010) co-located at ISWC 2010. 2010. Shangai, China.
- QUIRINO, G. K.; NARDI, J. C.; BARCELLOS, M. P.; et al. **Towards an Service Ontology Pattern Language**. Proceedings of the 34th International Conference on Conceptual Modeling (ER 2015). 2015. Stockolm, Sweden.
- RUY, F. B.; FALBO, R. DE A.; BARCELLOS, M. P.; GUIZZARDI, G. **Towards an ontology pattern language for harmonizing software process related ISO standards**. 30th Annual ACM Symposium on Applied Computing. p.388–395, 2015. ACM.
- RUY, F. B.; FALBO, R. DE A.; BARCELLOS, M. P.; GUIZZARDI, G.; QUIRINO, G. K. **An ISO-based software process ontology pattern language and its application for harmonizing standards**. ACM SIGAPP Applied Computing Review, v. 15, n. 2, p. 27–40, 2015.
- RUY, F. B.; REGINATO, C. C.; SANTOS, V. A.; FALBO, R. DE A.; GUIZZARDI, G. **Ontology Engineering by Combining Ontology Patterns**. 34th International Conference on Conceptual Modeling (ER'2015). p.173–186, 2015. Stockholm, Sweden: Springer.
- SALAH, D.; ZEID, A. **PLITS: A Pattern Language for Intelligent Tutoring Systems**. EuroPLOP. 2009.
- SANTOS, A. L.; KOSKIMIES, K. **Modular Hot Spots: A Pattern Language for Developing High-Level Framework Reuse Interfaces using Aspects**. EuroPLOP. 2008.

- SCHERP, A.; SAATHOFF, C.; FRANZ, T.; STAAB, S. **Designing core ontologies**. Applied Ontology, v. 6, p. 177–221, 2011.
- SESERA, L. **Applying fundamental banking patterns: stories and pattern sequences**. Proceedings of the 15th European Conference on Pattern Languages of Programs. 2010.
- TEIXEIRA, M. DAS G. DA S.; FALBO, R. DE A.; QUIRINO, G. K.; GUIZZARDI, G.; GAILLY, F.; BARCELLOS, M. P. **PoN-S: A Systematic Approach for Applying the Physics of Notation (PoN)**. Exploring Modelling Methods for Systems Analysis and Design Conference. 2016.
- TRAVASSOS, G.; GUROV, D.; AMARAL, E. **Introdução à Engenharia de Software Experimental**. Relatório Técnico ES-590/02 Programa de Engenharia de Sistemas e Computação COPPEUFRJ, p. 52, 2002.
- WANG, Y.; ZHAO, L.; WANG, X.; YANG, X.; SUPAKKUL, S. **PLANT: A pattern language for transforming scenarios into requirements models**. International Journal of Human-Computer Studies, v. 71, n. 11, p. 1026–1043, 2013.
- WARE, C. **Information Visualization, Second Edition: Perception for Design**. 2nd ed. Kaufmann, Morgan, 2004.
- WELLHAUSEN, T. **User interface design for searching: A pattern language**. Tenth European Conference on Pattern Languages of Programs. 2005. Irsee.
- WEYNS, D. **A pattern language for multi-agent systems**. Software Architecture, 2009 & European Conference on Software Architecture. WICSA/ECSA 2009. Joint Working IEEE/IFIP Conference on. p.191–200, 2009.
- WIERINGA, R.; MAIDEN, N.; MEAD, N.; ROLLAND, C. **Requirements engineering paper classification and evaluation criteria: a proposal and a discussion**. Requirements engineering, v. 11, n. 1, p. 102–107, 2005.
- WIERINGA, R.; MAIDEN, N.; MEAD, N.; ROLLAND, C. **Requirements engineering paper classification and evaluation criteria: a proposal and a discussion**. Requirements Engineering, v. 11, n. 1, p. 102–107, 2006.
- ZAMANI, B.; BUTLER, G.; KAYHANI, S. **Tool Support for Pattern Selection and Use**. Electronic Notes in Theoretical Computer Science, v. 233, p. 127–142, 2009.

- ZDUN, U. **Pattern language for the design of aspect languages and aspect composition frameworks**. Software, IEE Proceedings, v. 151, n. 2, p. 67–83, 2004a.
- ZDUN, U. **Supporting incremental and experimental software evolution by runtime method transformations**. Science of Computer Programming, v. 52, n. 1, p. 131–163, 2004b.
- ZDUN, U.; KIRCHER, M.; VOLTER, M. **Remoting patterns: Design reuse of distributed object middleware solutions**. Internet Computing, IEEE, v. 8, n. 6, p. 60–68, 2004.
- ZIANI, A.; HAMID, B.; GEISEL, J.; BRUEL, J. M. **A model-based repository of security and dependability patterns for trusted RCES**. Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on IEEE. p.448–457, 2013.

Apêndice A

Formulários Utilizados nos Estudos Experimentais

Este apêndice apresenta os formulários utilizados durante os estudos experimentais e survey realizados.

A1. Termo de Consentimento utilizado nos Estudos Experimentais

Estudo Experimental

Estudo empírico realizado na disciplina de Engenharia de Ontologias para análise do uso de uma OPL para criação de uma ontologia de domínio.

Termo de Consentimento

Termo de Consentimento do Estudo Experimental

 nemo	UFES (Universidade Federal do Espírito Santo) NEMO (Ontology & Conceptual Modeling Research Group) <i>Estudo empírico realizado da disciplina de Engenharia de Ontologias para análise do uso de uma OPL para criação de uma ontologia.</i>	 UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
--	---	--

Estudo Experimental: Análise do uso de uma OPL para criação de uma ontologia.

Este estudo tem o objetivo de analisar o uso de uma Linguagem de Padrões Ontológicos (OPL) na criação de ontologias. Ele servirá como uma avaliação preliminar da notação visual de OPLs e da aplicação do processo de OPLs no desenvolvimento de ontologias.

Após o aceite a este termo de consentimento, serão apresentadas algumas questões. É garantida a confidencialidade dos dados individuais cedidos no estudo. Os dados são destinados à realização da pesquisa, não sendo usados como avaliação pessoal ou profissional. É assegurado o anonimato dos participantes na publicação dos resultados da pesquisa.

Apesar de convidado, a participação é voluntária, sendo direito não querer participar ou abandonar a realização do estudo a qualquer momento.

Ao clicar em "Continuar", declaro ter mais de 18 anos de idade e participar voluntariamente da avaliação do estudo do uso de uma OPL para criação de uma ontologia, tendo lido as informações contidas neste termo antes de participar do estudo.

Continuar »

20% concluído

A2. Formulário de Caracterização utilizado nos Estudos Experimentais

Estudo Experimental

*Obrigatório

Perfil de Participante

Perfil de Participante de Estudo Empírico

Nome *

Informe seu nome completo

E-mail *

Informe o seu e-mail de contato

Formação acadêmica

Formação Acadêmica *

Informe o curso ou a área da maior titulação indicada

Situação Atual da Formação Acadêmica *

Escolha a sua situação atual com relação à Formação Acadêmica

- Completo
- Incompleto

Grau de Formação Acadêmica *

Escolha o seu grau de formação acadêmica entre as opções abaixo

- Superior
- Especialização
- Mestrado
- Doutorado

Experiência em Modelagem Conceitual

Grau de Experiência em Modelagem Conceitual *

Exemplo de experiências: uso de UML

- Nenhum
- Baixo (menos de 1 ano)
- Médio (entre 1 ano e 3 anos)
- Alto (mais de 3 anos)

Se possui experiência em Modelagem Conceitual, de que forma ela foi adquirida?

- Curso
- Disciplina
- Projeto (Profissional, Pesquisa, TCC, Extensão, Iniciação Científica ou similar)

Especifique abaixo o Curso, Disciplina e/ou Projeto que você indicou na questão anterior e a respectiva duração.

Ex: Projeto: 1 ano de Iniciação científica relacionada a integração semântica.

Experiência em Desenvolvimento de Ontologias

Experiência em Desenvolvimento de ontologias *

Exemplo de experiências: uso de alguma ontologia

- Primeira experiência nesta disciplina
- Baixa (menos de 1 ano)
- Média (entre 1 ano e 3 anos)
- Alta (mais de 3 anos)

Se possui experiência anterior no desenvolvimento de Ontologias, de que forma ela foi adquirida?

- Curso
- Disciplina
- Projeto (Profissional, Pesquisa, TCC, Extensão, Iniciação Científica ou similar)

Se possui experiência no desenvolvimento de ontologias, descreva sucintamente o(s) projeto(s) principal(is) desenvolvido(s) – projeto, contexto e atividades

Experiência em Linguagem de Padrões Ontológicos (OPL)

Experiência em Linguagem de Padrões Ontológicos (OPL) *

Exemplo de experiências: uso de alguma OPL

- Primeira experiência nesta disciplina
- Baixa (menos de 1 ano)
- Média (entre 1 ano e 3 anos)
- Alta (mais de 3 anos)

Se possui experiência em OPL, de que forma ela foi adquirida?

- Curso
- Disciplina
- Projeto (Profissional, Pesquisa, TCC, Extensão, Iniciação Científica ou similar)

Se possui experiência em OPL, descreva sucintamente o(s) projeto(s) principal(is) desenvolvido(s) – projeto, contexto e atividades

« Voltar

Continuar »



A3. Questionário utilizado nos Estudos Experimentais

Estudo Experimental

*Obrigatório

Questionário de Avaliação de Atividade

Responda as questões abaixo considerando sua percepção inicial da OPL.

1) Quão difícil/fácil você considera entender o mecanismo geral de funcionamento da OPL, que é composto por um processo e um conjunto de patterns? *

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Justificativa:

2) Quão difícil/fácil você considera entender cada uma das representações dos elementos da OPL:

Ponto de Entrada

 Entry Point

*

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Justificativa:

Padrão

Pattern

*

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Fluxo



- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Justificativa:

Grupo de Padrões



- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Justificativa:

Padrões variantes



- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Justificativa:

Nó de saídas alternativas e nó de junção de dependência múltipla.

- Fork Node: Parallel Output Flows (OR)
- Join Node: Parallel Input Flows (multiple dependency)

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Nó de decisão

 Decision Node: alternative paths

*

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Justificativa:

3) Dentre os elementos listados anteriormente, qual deles você teve maior dificuldade para entender? Justifique sua resposta. *

« Voltar

Continuar »

 60% concluído

4) Você teria alguma sugestão para melhorar a forma como os padrões são conectados, ou seja, para melhorar as notações visuais e símbolos da linguagem utilizados para representar o processo da OPL? *

5) Qual dos patterns da OPL você teve maior dificuldade para aplicar? Justifique sua resposta. *

6) A documentação fornecida (especificação da OPL e arquivo astah) atendeu suas necessidades para entendimento e utilização da OPL? *

- Atendeu Totalmente
- Atendeu Parcialmente
- Atendeu Muito Pouco
- Não Atendeu

7) Quais informações que, se disponíveis como parte da especificação da OPL, teriam sido úteis? *

« Voltar

Continuar »

80% concluído

Responda as questões abaixo considerando sua percepção sobre o processo de desenvolvimento de uma ontologia de domínio através do uso da OPL.

8) Qual o grau de dificuldade que você teve para utilizar a OPL para criar uma ontologia de domínio? Justifique sua resposta *

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

*

Justificativa:

9) Caso você já tenha tido uma experiência anterior com o desenvolvimento de ontologias, você considera que o processo de desenvolvimento da ontologia de domínio foi mais produtivo com o uso da OPL? Justifique sua resposta. (Caso esta seja sua primeira experiência no desenvolvimento de ontologias, desconsidere esta pergunta).

- Discordo Completamente
- Discordo
- Indiferente
- Concordo
- Concordo Completamente

Justificativa:

10) Quais dificuldades você teve para integrar os padrões utilizados? *

11) Você considera que o uso da OPL para desenvolver a ontologia de domínio contribuiu para a qualidade da ontologia resultante? Justifique sua resposta. *

- Discordo Completamente
- Discordo
- Indiferente
- Concordo
- Concordo Completamente

*

Justificativa:

« Voltar

Enviar

Nunca envie senhas pelo Formulários Google.

100% concluído.

A4. Questionário usado nas entrevistas dos estudos experimentais:

Entrevista

1) Descreva o processo geral que vocês seguiram para desenvolver a ontologia de domínio. Por onde vocês começaram? Em que momento começaram a usar a OPL?

2) Em relação às questões de competência que vocês definiram para a ontologia de domínio que desenvolveram:

- Como vocês definiram as QCs da ontologia de domínio? (*aguardar a respostas deles*)
- Vocês usaram (ou avaliaram) as QCs dos patterns da OPL? (*aguardar a respostas deles*).
- Se sim, como? Em que momento?

3) Em relação ao uso da OPL:

- Como foi feita a seleção dos patterns? (o que foi usado como base para selecioná-los)
- E a aplicação dos patterns e especialização dos conceitos? Vocês selecionaram todos os patterns e criaram um modelo a partir do qual os conceitos do domínio foram especializados ou para cada pattern selecionado vocês já foram especializando os conceitos do domínio?
- Como foi realizada a integração dos patterns? E a integração dos conceitos do domínio com os demais conceitos do modelo?

4) Vocês tiveram dificuldades para integrar (unir) os padrões utilizados? E para integrar os conceitos do domínio aos demais conceitos do modelo? Se sim, quais.

5) Como vocês exploraram/identificaram os conceitos específicos do domínio (conceitos que não estavam presentes na OPL)?

6) Vocês tiveram dificuldades em identificar os conceitos específicos do domínio (conceitos que não estavam presentes na OPL)?

Se sim:

- O que vocês acham que pode ter gerado essa dificuldade?
- Vocês acham que a dificuldade para identificar os conceitos específicos do domínio pode ser consequente do uso da OPL ou da pouca experiência com o domínio e com ontologias?
- Vocês acham que os conceitos da OPL inibiram vocês de identificar outros conceitos mais específicos do domínio? Se sim, por qual razão.

7) Dentre os elementos da OPL citados no questionário, os que geraram maior dúvida na interpretação foram: o Merge Node, Join/Fork Node e o Variant Patterns. O que vocês consideram que pode ter impulsionado essa dificuldade no entendimento desses elementos?

- Vocês acham que a dificuldade encontrada foi devido aos símbolos não serem claros ou se trata de desconhecimento dos símbolos da UML?

8) Qual dos patterns da OPL você teve maior dificuldade para aplicar?

9) Em relação aos estereótipos dos conceitos da OPL:

- Vocês consideraram os estereótipos ao especializarem os conceitos da ontologia de domínio?

- Vocês acham que se tivessem sido disponibilizados os patterns básicos de OntoUML (por exemplo, um pattern para modelar role mixins) alguns erros no modelo da ontologia de domínio poderiam ter sido evitados?

10) Vocês possuem alguma experiência com desenvolvimento de ontologias? Em caso afirmativo, você considera que o processo de desenvolvimento da ontologia de domínio foi mais produtivo com o uso da OPL?

11) Quais informações que vocês consideram que poderiam ter sido providas para facilitar a construção de uma ontologia através da utilização da OPL?

A5. Questionário utilizado no *Survey*.

Estudo Experimental

*Obrigatório

Perfil do Participante

Perfil de Participante de Estudo Empírico

Nome *

Informe seu nome completo

E-mail *

Informe seu e-mail de contato

Avaliação da Notação Visual de OPLs

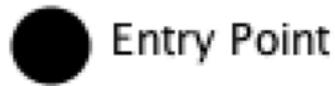
1) Quão difícil/fácil você considera entender o mecanismo geral de funcionamento da nova versão da OPL, que é composto por um processo (descrito em diferentes níveis de abstração), um conjunto de patterns e um modelo estrutural mostrando as relações entre os patterns? *

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

*

Justifique sua resposta.

2) Quão difícil/fácil você considera entender cada uma das representações dos elementos da OPL:



Ponto de Entrada *

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Justifique sua resposta



Padrão *

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Justifique sua resposta

→ Control Flows

Fluxo de Controle *

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Justifique sua resposta

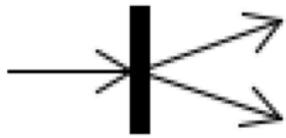


Decision Node:
alternative paths

Nó de Decisão *

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Justifique sua resposta

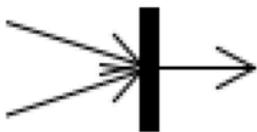


Fork Node: parallel paths

Nó de Bifurcação em Fluxos Paralelos *

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Justifique sua resposta



Join Node: junction of parallel paths

Nó de Junção de Fluxos Paralelos *

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil



Group

Grupo de Padrões *

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Justifique sua resposta



Variant
patterns

Grupo de Padrões Variantes *

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Justifique sua resposta



End Point

Nó de Finalização *

- Muito Difícil
- Difícil
- Neutro
- Fácil
- Muito Fácil

Justifique sua resposta

3) Dentre os elementos listados anteriormente, qual deles você teve maior dificuldade para entender? Justifique sua resposta. *

4) Você teria alguma sugestão para melhorar a forma como os padrões são conectados, ou seja, para melhorar as notações visuais e símbolos da linguagem utilizados para representar o processo da OPL? *

5) Você considera que essa versão tornou o processo mais claro? *

- Muito menos claro
- Menos claro
- Mantém a mesma clareza
- Mais claro
- Bem mais claro

*

Justifique sua resposta

6) Em relação à facilidade de entendimento do processo da OPL, qual representação você julga melhor? *

- A antiga, que permitia apenas mostrar o processo como um todo;
- A nova, que contempla diferentes níveis de abstração, permitindo trabalhar com uma visão geral do processo e com processos detalhados, que podem ser trabalhados no todo ou por grupos.

*

Justifique sua resposta

7) Em relação ao modelo que apresenta os componentes da S-OPL e as relações de dependência entre eles, você considera que ele facilita ou dificulta a aplicação e o entendimento da S-OPL? *

- Dificulta muito
 - Dificulta
 - Neutro
 - Facilita
 - Facilita muito
-

Justifique sua resposta

8) Você teria alguma sugestão para melhorar a forma como os componentes da OPL e as dependências entre eles são representadas, ou seja, para melhorar as notações visuais e símbolos da linguagem utilizados para representar a estrutura da OPL? *

« Voltar

Enviar



100% concluído.

Nunca envie senhas pelo Formulários Google.