

UNIVERSIDADE ESTADUAL DE MARINGÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

HENRIQUE VIGNANDO

**Uma Ontologia para Experimentos e Quasi-Experimentos
Controlados para Linha de Produto de Software -
OntoExper-LPS**

Maringá
2020

HENRIQUE VIGNANDO

**Uma Ontologia para Experimentos e Quasi-Experimentos
Controlados para Linha de Produto de Software -
OntoExper-LPS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Edson A. Oliveira
Junior

Maringá
2020

FOLHA DE APROVAÇÃO

HENRIQUE VIGNANDO

Uma Ontologia para Experimentos e Quasi-Experimentos Controlados para Linha de Produto de Software - OntoExper-LPS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação pela Comissão Julgadora composta pelos membros:

BANCA EXAMINADORA

Edson A. Oliveira Junior
Universidade Estadual de Maringá — DIN/UEM

Prof. Dr. Fulano 1
Universidade Estadual de Maringá — DIN/UEM

Prof. Dr. Fulano 2
Universidade Federal do Fim do Mundo — DEINFO/UFFM

Aprovada em: 21 de Fevereiro de 2019.

Local da defesa: Sala 101, Bloco C56, *campus* da Universidade Estadual de Maringá.

AGRADECIMENTOS

Agradeço a Deus por ter me abençoado nesta caminhada e as pessoas que contribuíram na realização deste trabalho. Em especial:

A minha família pelo carinho, apoio e incentivo.

Ao meu orientador professor Dr. xxxxxxxxxxxx pelo apoio, comentários e sugestões no desenvolvimento deste projeto.

Aos demais professores das disciplinas que cursei e aos colegas ...

E a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro concedido a este trabalho.

Uma Ontologia para Experimentos e Quasi-Experimentos Controlados para Linha de Produto de Software - OntoExper-LPS

RESUMO

O processo de experimentação em Engenharia de Software (ES) é fundamental para ciclo de vida de um software. Com ele é possível reduzir grandes esforços de desenvolvimento e principalmente de manutenção. A comunidade de ES vem discutindo e avaliando como melhorar a qualidade dos experimentos, visando aumentar a confiabilidade dos seus resultados. Por mais que eles tem abordado a qualidade de experimentos controlados de forma geral, ainda não há evidências que estão analisando em contextos específicos, como é o caso de Linhas de produto de Software (LPS). Neste caso, ainda existe uma falta de instrumentação e medição específica da qualidade dos experimentos em LPS. Por isso torna-se necessário fornecer um corpo de conhecimento confiável e replicável no contexto de LPS. Devido essa importância, projetar, executar e analisar os resultados de um experimento em LPS torna-se crucial para garantir a qualidade dos mesmos. Neste sentido propomos uma ontologia (OntoExper-LPS) para experimentos em LPS, pois possui centenas de experimentos publicados. A ontologia é concebida principalmente com base em diretrizes definidas e é projetada usando linguagem OWL, suportada pelo ambiente Protégé para verificação de sintaxe e avaliação inicial. A OntoExper-LPS foi preenchida com mais de 200 experimentos em linhas de produtos de software, reunidos em um estudo de mapeamento sistemático. Com este trabalho surge também a oportunidade de investigar a elaboração de um sistema de recomendação para experimentos em LPS se baseando sobre inferências aplicada OntoExper-LPS. Portanto, este trabalho apresenta conceitos fundamentais para elaboração de uma ontologia voltada para experimentos em LPS, e como prova de conceito, um sistema de recomendação para experimentos em LPS. Acreditamos que essa ontologia bem como o sistema de recomendação pode contribuir para documentar melhor os elementos essenciais de um experimento, promovendo assim a repetição, a replicação e a reprodutibilidade dos experimentos. Que por meio destes, possa levar qualidade para os projetos experimentais e resultados obtidos da modelagem ontológica e do sistema de recomendação. Apresentamos uma avaliação da OntoExper-LPS com relação a 8 critérios de qualidade, onde foi aplicado um questionário e enviado à especialistas da área. Os resultados dessa avaliação são apresentados em um capítulo a parte. Ao final temos um sistema de recomendação implementado como prova de conceito como uso de inferência na OntoExper-LPS, que apresenta bons resultados em recomendações para experimentos controlado utilizando

métodos híbridos de sistemas de recomendação. Espera-se com este projeto, contribuir com a comunidade de LPS no sentido de melhorar os projetos e execução de experimentos, aumentando a confiança do corpo de conhecimento visando a transferência de tecnologia para indústria.

Palavras-chave: Experimento Controlado de Software. Linha de Produto de Software. Ontologia. Ontologia em Engenharia de Software. Qualidade de Experimentos. Sistemas de Recomendação em Engenharia de Software. Sistemas de Recomendação.

ABSTRACT

The process of experimentation in Software Engineering (ES) is fundamental to the life cycle of a software. It is possible to reduce major development efforts and mainly maintenance. The ES community has been discussing and evaluating how to improve the quality of the experiments, in order to increase the reliability of its results. As much as they have approached the quality of generally controlled experiments, there is as yet no evidence they are analyzing in specific contexts, such as Software Product Lines (LPS). In this case, there is still a lack of instrumentation and specific measurement of the quality of experiments in LPS. It is therefore necessary to provide a reliable and replicable body of knowledge in the context of LPS. Because of this importance, designing, executing and analyzing the results of an experiment in LPS becomes crucial to guarantee the quality of the experiments. In this sense we propose an ontology (SMartyOntology) for experiments in LPS, because it has hundreds of published experiments. The ontology is primarily designed based on defined guidelines and is designed using OWL language, supported by the Protégé environment for syntax checking and initial evaluation. The ontology was filled with more than 150 experiments in software product lines, assembled in a systematic mapping study. It is also the opportunity to investigate the elaboration of a recommendation system for experiments in LPS based on the information modeling structured by the proposed ontology. Therefore, this paper presents fundamental concepts for the elaboration of an ontology for experiments in LPS, and for the creation of a recommendation system for experiments in LPS. We believe that this ontology as well as the recommendation system can contribute to better document the essential elements of an experiment, thus promoting the replication, replication and reproducibility of the experiments. In which, it can bring quality to the experimental projects and results obtained through the recommended experiment. Ontologies and Recommendation Systems are well known in ES, it is believed to be possible to apply these theories to recommend experiments in LPS. We also present a feasibility evaluation of the proposed ontology. At the end we have a recommendation system that shows good results in recommendations for controlled experiments. It is also hoped with this project to contribute to the LPS community in order to improve the projects and execution of experiments, increasing the trust of the body of knowledge in order to transfer technology to industry

Keywords: Controlled Software Experiment. Ontology. Quality of Experiments. Software Product Line. Systems of Recommendation in Software Engineering. Systems of Recommendation.

LISTA DE FIGURAS

Figura - 1.1	Etapas da Metodologia de Desenvolvimento de Pesquisa	17
Figura - 2.1	Um modelo de Características. Traduzido de Sommerville (2011)	22
Figura - 2.2	<i>Framework</i> de Engenharia de LPS (Pohl et al., 2005). Traduzido por Geraldi (2015)	23
Figura - 2.3	Conceitos Essenciais de um Experimento. Traduzido de Wohlin et al. (2012a)	25
Figura - 2.4	Visão Geral do Processo Experimental. Traduzido de Wohlin et al. (2012a)	26
Figura - 2.5	Estrutura de uma Ontologia. Autor	30
Figura - 2.6	Exemplo de uma Ontologia para o domínio: Destino de Viagem. Autor	31
Figura - 3.1	Initial graph of ontology.	36
Figura - 3.2	Conceptual Model Clustering.	37
Figura - 3.3	Class Diagram for proposal of the ontology model.	38
Figura - 3.4	Definição da classe Experiment no Protégé.	39
Figura - 3.5	Definição da propriedade de objeto documentation no Protégé. . .	40
Figura - 3.6	Definição da propriedade de dado nameSPLUsed no Protégé. . . .	41
Figura - 3.7	Grafo da modelagem da ontologia - gerado pelo WbVOWL. . . .	42
Figura - 3.8	Graph Path to Recommendation Example.	50
Figura - 5.1	Modelagem geral da RSSE proposta	65
Figura - 5.2	Passos de Construção para um RSSE. Traduzido e Adaptado de Maki (2016)	68

LISTA DE TABELAS

Tabela - 2.1	Definições dos conceitos de qualidade em experimentos em Engenharia de Software. Traduzido de Kitchenham et al. (2007)	27
Tabela - 3.1	Type of the Proposed Ontology	35
Tabela - 3.2	Classes and Sub-classes of ontology prototype	55
Tabela - 3.3	Result Summary of the OOPS! evaluation!	56

LISTA DE SIGLAS E ABREVIATURAS

ALM: *Application Lifecycle Management*

API: *Application Programming Interface*

AQ: Avaliação de Qualidade

CBF: *Content-based Filterin*

ES: Engenharia de Software

ETL: *Extract, Transform, Load*

GRSSE: Grupo de pesquisa em Reuso Sistemático de Software e Experimentação

ID: Identificador Único

LPS: Linha de Produto de Software

PDI: *Pentaho Data Integration*

POO: Programação Orientado a Objetos

RSSE: *Recommendation System in Software Engineering*

SUMÁRIO

1	Introdução	14
1.1	Contextualização	14
1.2	Motivação e Justificativa	16
1.3	Objetivos	16
1.4	Metodologia de Desenvolvimento	17
1.5	Organização do texto	18
2	Fundamentação Teórica	20
2.1	Considerações Iniciais	20
2.2	Linha de Produto de Software	20
2.3	Experimentos e <i>Quasi</i> -Experimento em Engenharia de Software	24
2.4	Qualidade de Experimentos em Engenharia de Software	27
2.5	Ontologia	28
2.6	Trabalhos Relacionados	29
2.7	Considerações Finais	32
3	Uma Ontologia para Experimentos em LPS - SMartOntology	34
3.1	Considerações Iniciais	34
3.2	Concepção	34
3.3	Projeto	38
3.3.1	Modelagem com Protegé	38
3.3.2	Povoamento com Pyton	40
3.4	Exemplo de Aplicação	49
3.5	Avaliação Empírica	51
3.6	Considerações Finais	54
4	Avaliação da SMartOntology: Um estudo Quantitativo	57
4.1	Considerações Iniciais	57
4.2	Escala Likert	59
4.3	Definição do Estudo	60
4.4	Planejamento do Estudo	60
4.5	Execução do Estudo	61
4.6	Análise e Discussão dos Resultados	61
4.6.1	Perfil dos Especialistas	62
4.6.2	Frequência Likert	62

4.6.3	Teste de Normalidade	62
4.6.4	Teste Estatístico	62
4.6.5	Correlação dos Critérios	62
4.6.6	Melhorias	62
4.7	Ameaças à Validade	62
4.7.1	Ameaças à Validade Interna	62
4.7.2	Ameaças à Validade Externa	63
4.7.3	Ameaças à Validade de <i>Constructo</i>	63
4.7.4	Ameaças à Validade de Conclusão	63
4.8	Empacotamento e Compartilhamento	64
4.9	Propostas de Melhoria para SMartOntology	64
4.10	Considerações Finais	64
5	Um Sistema de Recomendação para Experimentos em LPS (Prova de Conceito) - RecSysExper-SPL	65
5.1	Considerações Iniciais	65
5.2	Sistema de Recomendação	66
5.3	Concepção	69
5.4	Projeto	69
5.5	Ambiente de Desenvolvimento	69
5.6	Interface com Usuário	69
5.7	Considerações Finais	69
6	Conclusão	70
6.1	Considerações Iniciais	70
6.2	Considerações Finais	71
	REFERÊNCIAS	72
A	Apêndice A: Código Fonte do Modelo da SMartOntology	76
A.1	Saída Da Opção TIME-FLAGS com os <i>Breakpoints</i> .dijkstra e .main . .	76
B	Apêndice B: Código Fonte do Povoamento de Dados no Modelo SMartOntology	77
C	Apêndice C: Questionário de Avaliação e Respostas Aplicado à SMartOntology	78

Introdução

1.1 Contextualização

Foi discutido que a experimentação em Engenharia de Software (ES) tem se tornado fundamental para desenvolver e melhorar métodos e ferramentas, bem como melhorar os processos de manutenção de software (Kitchenham et al., 2007). Essa discussão permite que o conhecimento seja gerado de forma sistemática, disciplinada, quantificável e controlada (Wohlin et al., 2012a). Dessa forma melhora-se a qualidade dos experimentos¹ que por sua vez vem construir um corpo de conhecimento confiável e referente na área de experimentação em ES.

Atualmente, não há diretrizes específicas para avaliar a qualidade de experimentos em ES, especialmente, para áreas emergentes e em processo de consolidação como é o caso de Linha de Produto de Software (LPS), em que aspectos específicos do domínio como, por exemplo, os artefatos utilizados como objetos experimentais, a complexidade do treinamento, a dificuldade de seleção de participantes qualificados e a falta de repositórios, podem influenciar os experimentos. Além disso, tem-se percebido uma constante carência de documentação adequada dos experimentos que acabam por inviabilizar a repetição e auditoria dos estudos em LPS.

Ontologias estão entre os métodos mais utilizados para formalizar informações. Ontologias são representações formais de uma abstração contendo definições formais de nomenclatura, conceitos, propriedades e relações entre os conceitos, a fim de definir

¹Neste trabalho usaremos o termo "experimento" para denotar ambos os conceitos de "experimento controlado" e "quasi-experimento controlado".

um vocabulário controlado de termos e relações de conceitos [referência]. Assim, o uso de ontologias para representar informações sobre experimentos, padroniza os dados facilitando a interoperabilidade, a troca de informações e a replicação de experimentos.

Por mais atraente que seja, uma solução de ontologia para toda ES é muito esparsa. Dadas as particularidades de cada área em ES, seria complexo projetar uma ontologia capaz de representar todas as características particular de cada área em um único trabalho. Dessa forma, concentramos nossos esforços no campo da LPS, devido à nossa experiência em grupo, uma vez que um mapeamento sistemático foi realizado em centenas de experimentos publicados em LPS citeFurtado2018. (Grupo de Pesquisa em Reutilização Sistemática de Software e Experimentação Contínua - GReater).

Uma LPS é determinado por um conjunto e produtos de um determinado segmento de mercado (Northrop et al., 2007), como um sistema para aparelhos celulares, onde há ativos centrais com as principais funcionalidades que o software deve implementar, chamadas semelhanças, e pode haver um conjunto de funcionalidades específicas para determinados dispositivos, chamados variabilidades. Experimentos no campo de LPS requerem considerável experiência em ES, pois durante o planejamento e execução dos experimentos vários artefatos específicos de ES e LPS podem ser gerados. Assim, a curva de aprendizado é longa o suficiente para extrair e apresentar os resultados de experimentos de LPS satisfatoriamente.

Portanto, a definição de uma ontologia para experimentos específicos em LPS, se destina a auxiliar outros pesquisadores no desenvolvimento e replicação de novos experimentos, além de auxiliar na auditoria e validação de experimentos baseados em diretrizes determinadas pela ontologia.

Além disso, dados formalmente representados permitem o desenvolvimento de sistemas especialistas, como sistemas de recomendação. Um sistema de recomendação para experimentação ES poderia ser útil de duas maneiras: (i) didaticamente, para ajudar os alunos e professores, pesquisar experimentos relacionados; e, (ii) ajudar os profissionais da Engenharia de Software Experimental (ESE) a planejar e executar um experimento seguindo a experiência de outros, se baseando em consultas e recomendações de experimentos correlacionadas a sua área de atuação.

Portanto, o objetivo deste trabalho de mestrado é propor uma ontologia para representar formalmente essa diretriz, a fim de padronizar os experimentos de ES em SPL. Esperamos facilitar a representação dos dados de experimentos para aumentar o uso geral de experimentação em ES, aumentar o número de replicações e o compartilhamento de dados.

Resultados obtidos com a avaliação da qualidade e viabilidade da ontologia indicam a possibilidade de criar inferências aos dados e diretrizes sobre os experimentos, dessa forma é possível criar como prova de conceito, modelos de inferência para um sistema de recomendação sobre esta proposta de ontologia.

1.2 Motivação e Justificativa

Realizar um experimento em LPS exige alguns pontos de atenção específicos para garantir a qualidade do experimento. Estes pontos foram investigados em um trabalho de mestrado do nosso - GReater. Neste trabalho foi elaborado diretrizes para a determinar a qualidade de experimentos em LPS. Esta tarefa possui um árduo trabalho para garantir que, aspectos específicos do domínio como, por exemplo, os artefatos utilizados, que são, os objetos experimentais, a complexidade do treinamento, a dificuldade de seleção de participantes qualificados em LPS e a falta de repositórios de LPS, não influenciam nos experimentos ao ponto de invalidá-los. A falta de experimentos com qualidade afeta diretamente a possibilidade de repetição dos estudos em LPS.

Sabendo que para realizar um experimento em LPS com qualidade exige-se seguir alguns modelos e diretrizes, construir uma modelagem formal por meio de uma ontologia para representação desse conhecimento adquirido, pode proporcionar facilidade ao desenvolvimento dos mesmos, incentivando a cultura e desenvolvimento de experimentos na academia e indústria, seguindo um modelo formal de estrutura do conhecimento sobre experimento em LPS.

Por meio do GReater, foi encontrada uma lacuna nas pesquisas de qualidade em experimentos de ES em LPS que proporciona esta pesquisa, apresentando um campo aberto à pesquisa para determinar qualidade e modelagem formal de experimentos em LPS.

1.3 Objetivos

Esta pesquisa tem como objetivo geral, modelar uma ontologia que representa formalmente o conhecimento adquirido até o estado da arte sobre experimentos de ES em LPS. Chamaremos essa proposta de **OntoExper-LPS**, uma ontologia para apoiar experimentos em LPS.

Os objetivos específicos deste projeto são:

- gerar e representar um conjunto de metadados a partir das informações sobre experimentos em LPS;
- definir um modelo de ontologia para experimento de ES em LPS;
- provar ontologia com experimentos de ES em LPS;
- avaliar a qualidade e viabilidade do modelo proposto;
- elaborar um sistema de recomendação como prova de conceito do modelo de ontologia; e
- avaliar e empacotar o modelo, os resultados da avaliação e o sistema de recomendação.

1.4 Metodologia de Desenvolvimento

Para o desenvolvimento deste trabalho, foi necessário uma pesquisa exploratória para definir um modelo de ontologia para experimentos de ES em LPS. Para tal resultado, foi realizada uma pesquisa não sistemática sobre ontologias para experimentos, foi desenvolvido um projeto de ontologia para OntoExper-LPS, foi criado um programa automatizado para povoar a mesma, foi realizada uma avaliação da qualidade e viabilidade do modelo, e em seguida, foi elaborado um sistema de recomendação como prova de conceito. A Figura 1.1 apresenta as etapas da metodologia utilizada neste trabalho, descritas a seguir.

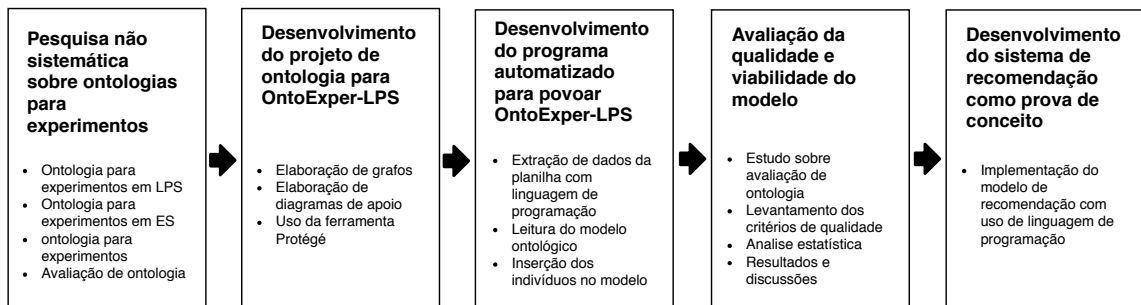


Figura 1.1: Etapas da Metodologia de Desenvolvimento de Pesquisa

- **Pesquisa não sistemática sobre ontologias para experimentos:** nesse momento foi realizado estudos não sistemático para encontrar trabalhos relacionados descrevendo ontologias para experimentos de ES em LPS e avaliação de ontologia, foram encontrado alguns estudos que serviram de apoio a essa pesquisa, eles estão descritos na Seção 2.6 do Capítulo 2;

- **Desenvolvimento do projeto de ontologia para OntoExper-LPS:** trata sobre desenvolvimento da OntoExper-LPS, que incluiu realizar a escolha das tecnologias e ferramenta de construção de ontologia, como por exemplo o Protégé, envolver os *stakeholders* no projeto com experiência no domínio de experimentação, selecionar abordagens e modelos de ontologias relacionados, prototipação e diagramação do projeto;
- **Desenvolvimento do programa automatizado para povoar OntoExper-LPS:** refere-se ao processo de desenvolvimento do programa automatizado de povoamento da ontologia, para isso foi necessário o uso de ferramentas, bibliotecas e linguagens de programação, foram mais de 200 experimentos encontrado na literatura, e em seguida foi executado o programa de extração de dados, povoando a ontologia por meio dos metadados mais o programa de extração;
- **Avaliação da qualidade e viabilidade do modelo:** refere-se ao estudo quantitativo, com objetivo de avaliar a qualidade da OntoExper-LPS, para isso foi elaborado um questionários com oito critérios de qualidade, enviado para os especialistas responderem, e posteriormente realizado uma análise das respostas, bem como pontos de melhorias encontrado durante a avaliação.
- **Desenvolvimento do sistema de recomendação como prova de conceito:** após processar os indivíduos na OntoExper-LPS, foi possível elaborar como prova de conceito, um sistema de recomendação utilizando o modelos de recomendação *Collaborative Filtering*, além de *frameworks* e linguagens de programação.

1.5 Organização do texto

Este capítulo apresentou a contextualização desta dissertação, a motivação e a justificativa, objetivos e metodologia. O restante da dissertação está estruturado da seguinte forma: o Capítulo 2 apresenta a Fundamentação Teórica sobre LPS, experimentais e quasi-experimentos em ES, qualidade de experimentos e quasi-experimentos em ES, Ontologias e Sistemas de Recomendação; o Capítulo 3 apresenta uma Ontologia para Experimentos em LPS - OntoExper-LPS, destacando desde a concepção ao povoamento da ontologia proposta e uma pré-análise de pontos de falha da ontologia; o Capítulo 4 apresenta uma avaliação quantitativa sobre a qualidade da OntoExper-LPS por especialistas da área; o Capítulo 5 apresenta um sistema de recomendação para experimentos em LPS, usando como modelo de dados a OntoExper-LPS; e o Capítulo 6 apresenta as

considerações finais acerca deste projeto de dissertação de mestrado, bom como as suas contribuições, limitações e trabalhos futuros.

Fundamentação Teórica

2.1 Considerações Iniciais

Este capítulo apresenta conceitos importantes sobre a fundamentação teórica, necessárias para a compreensão desta pesquisa como LPS, experimento e *quasi*-experimento em ES, qualidade de experimentos e *quasi*-experimento em ES, ontologia, sistemas de recomendação e trabalhos relacionados. Dessa forma a estrutura deste capítulo fica, na Seção 2.2 apresenta os principais pontos de LPS e variabilidades; na Seção 2.3 é apresentado uma visão geral de experimento e *quasi*-experimento em ES; na Seção 2.4 são apresentados informações preliminares sobre qualidade de experimentos e *quasi*-experimento em ES; A Seção 2.5 descreve os fundamentos sobre o que é uma ontologia e sua finalidade; Seção 2.6 são discutidos os trabalhos relacionados; e na Seção 2.7 são apresentadas as Considerações finais deste capítulo.

2.2 Linha de Produto de Software

Uma Linha de Produto de Software (LPS) é um conjunto de produtos que endereçam a um determinado segmento de mercado ou missão particular (Northrop et al., 2007). Esse conjunto de produtos também é denominado família de produtos, no qual os membros desta família são produtos específicos gerados a partir da reutilização de uma infraestrutura comum, denominada núcleo de artefatos (*Core assets*).

O núcleo de artefatos é composto por um conjunto de características comuns chamadas de similaridades, e características variáveis chamadas de variabilidades (Van der Linden

et al., 2007). Este núcleo forma a base da LPS que determina a Arquitetura de uma LPS, que são eles, componentes reusáveis, modelos de domínios, requisitos da LPS, planos de testes e modelos de características de variabilidades.

O modelo de características contém todas as características de uma LPS e os seus inter-relacionamentos. De acordo com Apel, "uma característica é um comportamento característico ou visível ao usuário final de um sistema de software". Uma característica pode ser obrigatória, opcional ou alternativa. O modelo de características representa as variabilidades e as variantes de uma LPS (Apel, 2013).

- Variabilidades são descritas por: ponto de variação que permite a resolução de variabilidades em artefatos genéricos de uma LPS, e;
- Variantes são representadas pelos: possessíveis elementos que podem ser escolhidos para resolver um ponto de variação.

Restrições entre variantes, estabelecem os relacionamentos entre uma ou mais variantes, com o objetivo de resolver seus respectivos pontos de variação ou variabilidade em um dado tempo de resolução (Halmans e Pohl, 2003).

O *MobileMedia* é um exemplo didático de LPS, o produto final é um software que gerencia as mídias de um aparelho celular. O núcleo de artefatos deve conter algumas das seguintes características, opções de criar, visualizar, remover e editar a legenda da imagem. As características variantes opcionais podem ser, a capturar uma nova imagem, ordenar e favoritar imagens. As características variantes alternativas podem ser os diversos tamanhos de tela. O produto final deve possuir ao menos uma variante alternativa.

A Figura - 2.1 apresenta um grafo com o Modelo de Características da LPS *MobileMedia*. As arestas com círculos preenchido representa as características pertencentes ao *core asset*. As arestas com círculos vazios representam características opcionais. As arestas ligadas por um triângulo, como as que saem do vértice Seleção de Mídia, representam características alternativas, por exemplo, uma instância desta LPS deve possuir ao menos um tipo de seleção de mídia, seja ele, por Foto, Música ou Vídeo.

Uma instância deste exemplo teria as seguintes características no seu produto de software final:

- Gerenciamento de Álbum;
 - Criar Álbum;
 - Excluir Album;
- Gerenciamento de Mídia;

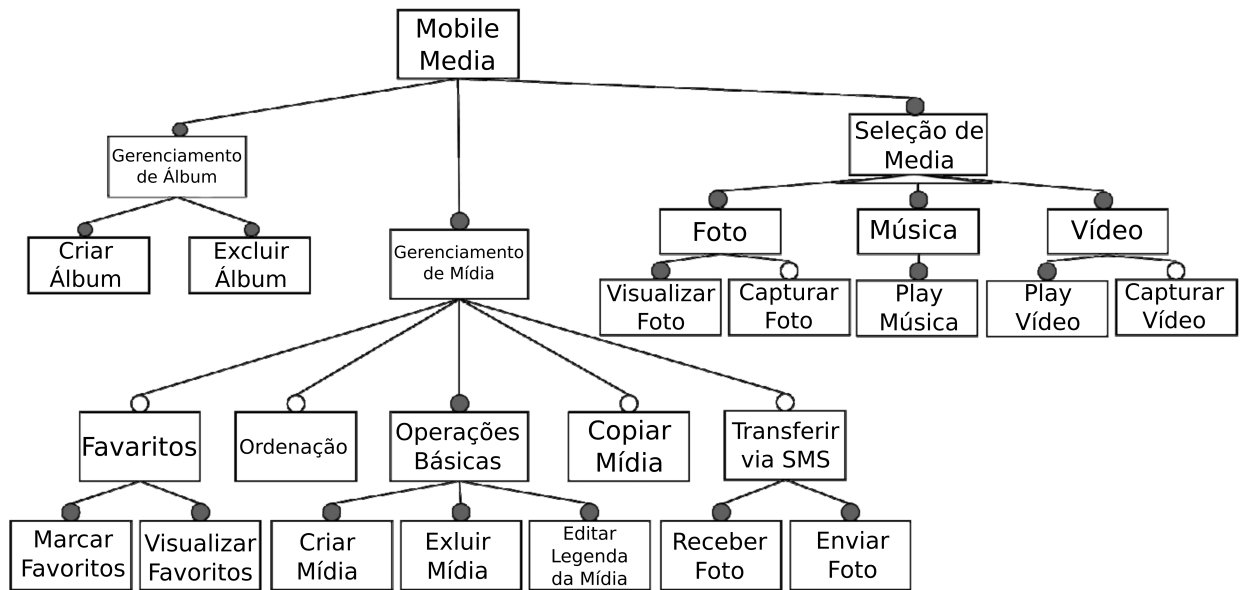


Figura 2.1: Um modelo de Características. Traduzido de Sommerville (2011)

- Operações Básicas
 - Criar Mídia
 - Excluir Mídia
 - Editar Legenda Mídia
- Favoritos;
 - Marcar Favoritos
 - Visualizar Favoritos
- Seleção de Mídia;
 - Foto
 - Visualizar Foto
 - Capturar Foto

Pohl, Böckle e van Der Linden desenvolveram o *framework* para engenharia de LPS. O objetivo deste de *framework* é incorporar os conceitos centrais da engenharia de linha de produto tradicional, proporcionando a reutilização de artefatos e a customização em massa por meio de variabilidades. O *framework* está dividido em dois processos, o de Engenharia de Domínio e o de Engenharia de Aplicação, conforme apresentado na Figura - 2.2.

- **Engenharia de Domínio:** processo em que as similaridades e as variabilidades das LPSs são identificadas e realizadas. No qual, é composto de cinco subprocessos principais, sendo eles: Gerenciamento de Produto, Engenharia de Requisitos do Domínio, Projeto do Domínio, Realização do Domínio e Teste de Domínio;
- **Engenharia de Aplicação:** processo em que as aplicações de uma LPS são construídas por meio da reutilização de artefatos de domínio, explorando as variabilidades de uma linha de produto. No qual, é composto pelos subprocessos: Engenharia de Requisitos da Aplicação, Projeto da Aplicação, Realização da Aplicação e Teste da Aplicação.

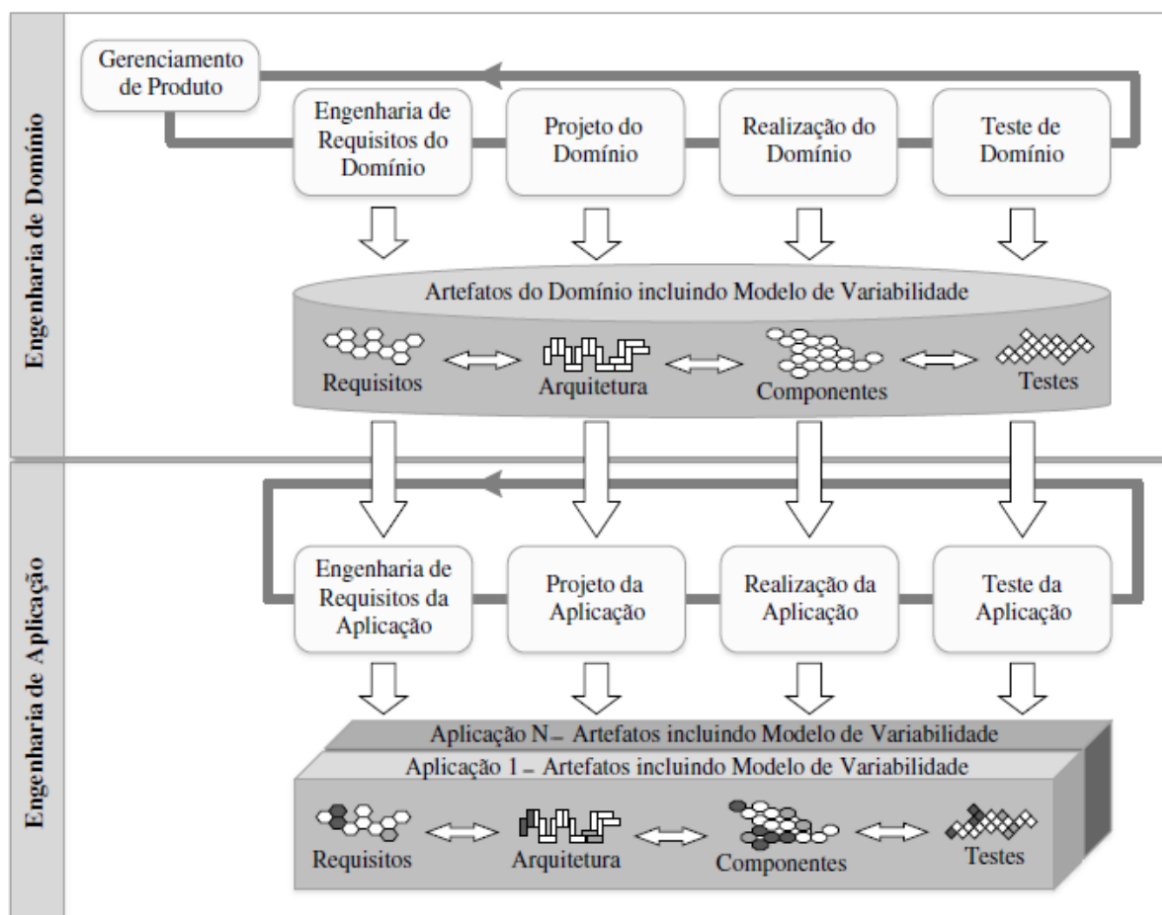


Figura 2.2: *Framework* de Engenharia de LPS (Pohl et al., 2005). Traduzido por Geraldi (2015)

2.3 Experimentos e Quasi-Experimento em Engenharia de Software

Existe uma diferença relevante entre experimento e *quasi*-experimento, esta diferença está relacionada a amostra do experimento. Quando se trata de um experimento a amostra é uma representação aleatória e válida de uma determinada população, ou seja, a amostra é uma representação da população. Quando se trata de *quasi*-experimento a amostra não é aleatória e não representa sua população. É difícil realizar experimentos em LPS, devido a dificuldade de determinar uma amostra representativa e aleatória da população, pois normalmente estas amostras são pessoas (Wohlin et al., 2012a).

Por meio de um modelo teórico entre dois ou mais fenômenos relacionados a fim de determinar se este modelo proposto pode ser considerado correto, se desenvolve o experimento onde relacionamos a causa e o efeito deste modelo. Assim, utiliza-se o modelo para criar uma hipótese em relação às mudanças particulares nos fenômenos (a causa) que levarão a mudanças no outro (o efeito). Logo, o papel do experimento é testar a hipótese para decidir se é verdadeira ou falsa (Kitchenham et al., 2015). A Figura - 2.3 apresenta à ideia de uma relação causa e efeito em teoria, na qual a parte superior à linha tracejada se encontra a teoria e, na parte inferior a observação (Wohlin et al., 2012a).

Um dos principais elementos de um experimento são as variáveis dependentes e independentes.

- **Variáveis independentes:** estão associadas à causa e controladas como resultado das atividades do experimentador, também são chamadas de fatores que podem assumir valores denominados tratamentos;
- **Variáveis dependentes:** estão associadas ao efeito e resultam nas mudanças que o experimentador realiza nas variáveis independentes (Kitchenham et al., 2015).

Segundo Kitchenham et al. 2015, existe uma característica dita fator de confusão em experimentos de ES que envolvem seres humanos. Esse fator pode ser representado pela presença de algum elemento indesejável no estudo que dificulta distinguir entre duas ou mais causas possíveis de um efeito que foi medido pela variável dependente como, por exemplo, os níveis de habilidade dos participantes e a extensão de suas experiências anteriores com o objeto experimental.

Em Engenharia de Software, especialmente em LPS, é difícil de se executar experimentos dado que estes devem possuir aleatoriedade completa em suas variáveis. Isto se deve à dificuldade de alocar os participantes e/ou objetos a diferentes tratamentos de

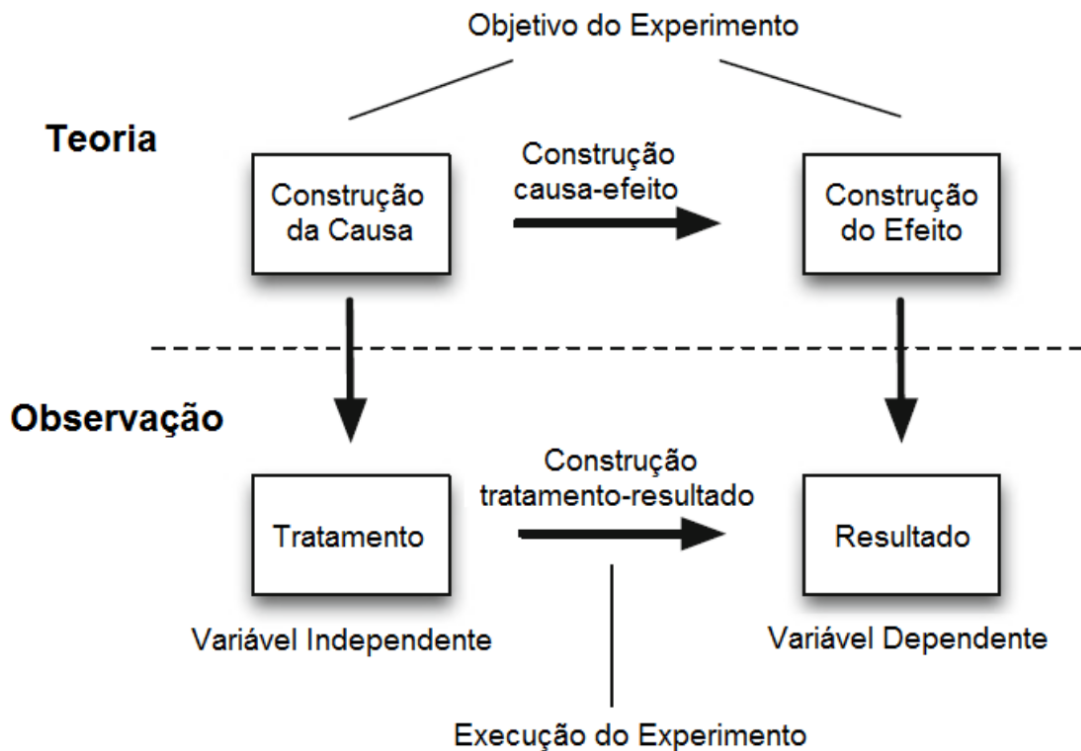


Figura 2.3: Conceitos Essenciais de um Experimento. Traduzido de Wohlin et al. (2012a)

maneira aleatória, bem como, à falta de representatividade do número de participantes em uma amostra da população. Portanto, os experimentos realizados nesta área são, frequentemente, *quasi*-experimentos, nos quais não há aleatoriedade dos participantes e/ou dos objetos experimentais, em ES chamados de artefatos de software, podendo ser processos ou ferramentas (Wohlin et al., 2012a).

Segundo Wohlin et al. 2012a a realização de um experimento pode ser dividido em um processo contendo cinco atividades, conforme apresentadas na Figura - 2.4 e descritas a seguir:

- **Definição:** é a primeira atividade, onde define-se o problema, objetivo e metas do experimento. Caso não seja devidamente estabelecida, pode ocorrer retrabalho ou o experimento não pode ser utilizado para se estudar o que era almejado;
- **Planejamento:** é uma preparação de como o experimento será conduzido, em que ocorre a determinação do contexto do experimento, a formulação das hipóteses, sendo **hipótese nula** que o experimentador espera rejeitar com a maior confiança possível e a **hipótese alternativa** que se espera aceitar, a seleção de variáveis (dependentes e independentes), a seleção dos participantes, o projeto

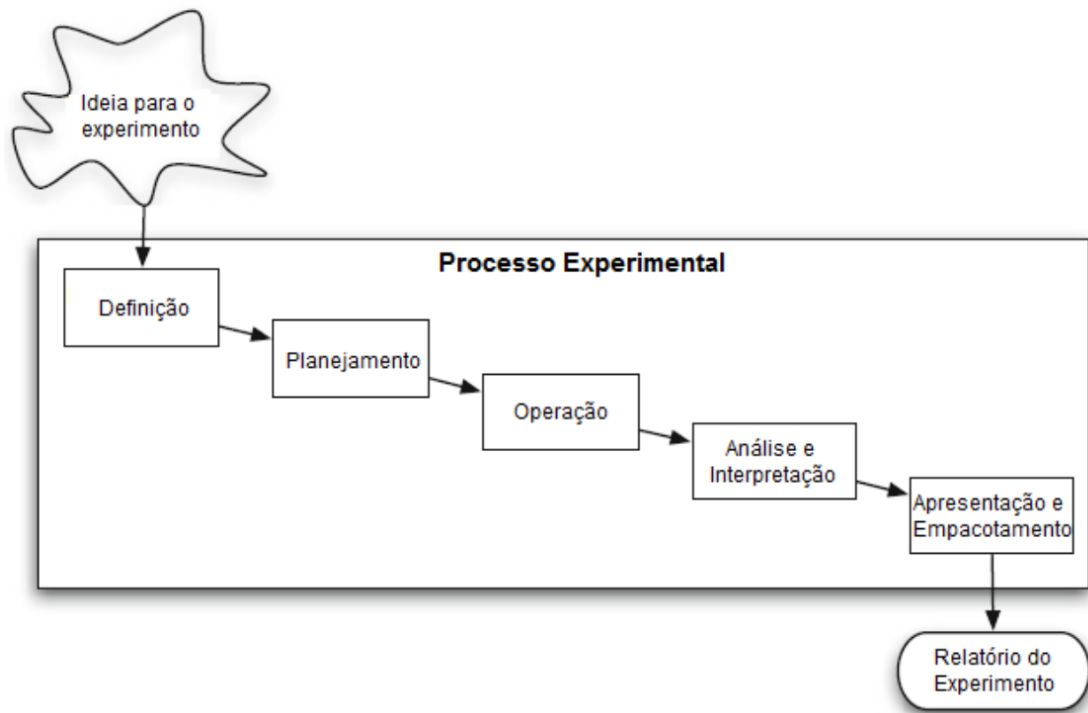


Figura 2.4: Visão Geral do Processo Experimental. Traduzido de Wohlin et al. (2012a)

do experimento, a instrumentação e a avaliação da validade, dividida em quatro tipos, sendo **validade interna** refere-se ao relacionamento tratamento-resultado; **validade externa** apresenta a generalização dos resultados a uma população maior; **validade de constructo** demonstra a relação entre a teoria e observação; e **validade de conclusão** refere-se a como os experimentadores foram aptos de analisar os resultados de um estudo e se a forma como foi feita é apropriada (Kitchenham et al., 2015).

- **Operação:** essa atividade é composta da **preparação** dos participantes e dos materiais necessários (instrumentação); **execução** das tarefas pelos participantes de acordo com diferentes tratamentos e coleta dos dados; e **validação dos dados** pelo experimentador, verificando os dados informados pelos participantes, de forma que os resultados do experimento sejam válidos;
- **Análise e Interpretação:** os dados coletados na atividade anterior são analisados utilizando a estatística descritiva. Após isso, é verificada a necessidade de redução do conjunto de dados, de forma a garantir que os dados representam uma informação correta e/ou esperada. Por fim, realiza-se o teste de hipóteses para avaliar estatisticamente se hipótese nula pôde ser rejeitada.

- **Apresentação e Empacotamento:** nessa atividade, os resultados são reportados, por exemplo, como artigos em conferência e/ou periódico, relatórios de tomada de decisão e empacotados para permitir a replicação do experimento, como material educativo, entre outros.

2.4 Qualidade de Experimentos em Engenharia de Software

Segundo Dieste e Juristo (2013), o conceito de qualidade de experimentos em ES pode ser visto em dois pontos de vista diferentes, o primeiro é considerar a qualidade como o resultado da validade interna de um bom experimento e o segundo é tornar a qualidade operacional assim como a quantidade de vieses nos resultados experimentais. Outro ponto colocado por Dieste e Juristo, é a validade externa que também tem uma função chave ao analisar se um experimento tem boa qualidade, porém essa função é contrária à validade interna.

Na Tabela - 2.1 são apresentadas as definições dos conceitos de qualidade citados anteriormente.

Tabela 2.1: Definições dos conceitos de qualidade em experimentos em Engenharia de Software. Traduzido de Kitchenham et al. (2007)

Termo	Sinônimo	Definição
Viés	Erro sistemático	Uma tendência para produzir resultados que partem sistematicamente de resultados "verdadeiros". Resultados sem viés são válidos internamente.
Validade Interna	Validade	O alcance em que o projeto e a condução do estudo são possíveis de evitar erro sistemático. A validade interna é um pré-requisito para a validade externa.
Validade Externa	Generabilidade, Aplicabilidade	O alcance em que os efeitos observados no estudo são aplicáveis fora do estudo.

Segundo Dieste e Juristo (2011) os experimentos de boa qualidade são aqueles livres de vieses. O viés está relacionado com a validade interna, por exemplo, quão bem os experimentos são planejados, executados e analisados (Dieste e Juristo, 2013). Para minimizar os vieses existem alguns métodos como, aleatorização para criar grupos experimentais homogêneos, imparcialidade para alocar os indivíduos. Desta forma os resultados podem

ser analisados mesmo depois do experimento ter sido realizado com replicações. Enquanto os experimentos de baixa qualidade seriam os que usam pouco ou nenhum dos métodos citados (Dieste e Juristo, 2011).

Como o viés não pode ser medido, existem algumas abordagens para avaliá-lo. Os instrumentos de Avaliação de Qualidade (AQ), são projetados para avaliar a validade interna e inferir a qualidade de experimentos, tais como, abordagens simples (questionários), *checklists* (contém ou não contém), escalas de qualidade, opinião de especialista (Dieste e Juristo, 2011, 2013; Teixeira, 2014).

Por outro lado a qualidade de um experimento em ES também pode ser avaliada considerando o projeto e análise dos experimentos, em termos de poder estatístico, análise do tamanho de efeito (resultado), *quasi*-experimentais e relatório de experimento (Kampenes, 2007).

Até o momento não se tem conhecimento de trabalhos que tratam a qualidade de experimentos em área específica de ES. Entretanto, foram recuperados alguns trabalhos, por meio de pesquisas não sistemáticas, que estão relacionados com a avaliação de qualidade dos experimentos em Engenharia de Software, descritos a seguir:

- Kitchenham et al. (2007) propõem um *checklist* de avaliação de qualidade de experimentos em ES contendo cinquenta questões para avaliar a qualidade de experimentos, em que sugerem que os pesquisadores selecionam apenas as questões do *checklist* mais adequadas ao contexto de suas próprias questões de pesquisa.
- Kitchenham et al. (2015) apresentam um *checklist* de avaliação de qualidade de experimentos em ES com nove questões, onde cada questão possui subquestões categorizadas em: (i) "Questões sobre objetivo", (ii) "Questões sobre o projeto, coleta de dados e análise do dados" e (iii) "Questões sobre o resultado do estudo".
- Dieste e Juristo (2011) desenvolveram uma escala de qualidade para determinar a qualidade de experimentos, contendo dez questões baseadas nas cinco dimensões de Kitchenham et al. (2007), sendo: contexto experimental, projeto experimental, análise, interpretação dos resultados e apresentação dos resultados. As respostas de cada questão são "sim" ou "não".

2.5 Ontologia

A palavra ontologia é formada por meio dos termos gregos *ontos* (ser) e *logos* (estudo, discurso), que engloba algumas questões abstratas como a existência de determinadas

entidades, o que se pode dizer que existe, qual o significado do ser, etc. Segundo Wolff e École (1962), ontologia é um ramo da filosofia que estuda a realidade e existência, ou o ser enquanto ser. Em outras palavras, é o estudo da descrição de coisas do mundo real. Outro ponto de vista proposto por Gruber (1993), diz que ontologias são uma especificação formal de uma contextualização e uma contextualização é uma visão abstrata e simplificada do mundo.

Ontologia em Computação, Sistemas de Informação e Ciência da Informação, é definida como um modelo de dados que representa um conjunto de conceitos dentro de um domínio e os relacionamentos entre estes. Uma ontologia é utilizada para realizar inferência sobre os objetos do domínio. No cenário atual, as ontologias em ciências da informação são utilizadas como uma forma de representação de conhecimento lógico, possibilitando a inferência de novos fatos com base nos dados armazenados na ontologia.

Uma ontologia define primitivas/diretrizes de um domínio de conhecimento, estas primitivas/diretrizes podem ser definidas como classes, atributos, propriedades e restrições. Essas definições seguem o padrão de representação conhecido como lógica descritiva. A lógica descritiva representa os conceitos de um domínio (chamado de **TBox** - *Terminological Box*) separadamente dos indivíduos (chamado de **ABox** - *Assertion Box*) (Calvanese et al., 2005). A lógica descritiva é mais representativa e eficiente que a lógica proposicional e a lógica de predicados (usados em linguagens de programação lógica, como Prolog).

Portanto uma ontologia para a representação de um conhecimento possui a seguinte estrutura: Uma base de conhecimento onde estão os dois conjuntos de conhecimento terminológico (**TBox**) e o conjunto de conhecimento sobre objetos (**ABox**), seguido de um mecanismo de inferência e uma aplicação para atuar na manipulação de informações extraídas do mecanismo de inferência, A Figura - 2.5 apresenta essa estrutura.

A Figura - 2.6 apresenta um exemplo de ontologia, por meio de um grafo, para o domínio: "destino de viagem". Os vértices ovais representam as classes, e os vértices retangulares representam os indivíduos (instâncias da classe). As arestas comuns representam um relacionamento de classe e subclasse, as arestas tracejadas representam um relacionamento de propriedade, já as arestas que começam com um losango indica a definição de uma propriedade, especificando sua tipagem.

2.6 Trabalhos Relacionados

Durante uma revisão não sistemática (*ad hoc*) da literatura foram encontrados alguns estudos que propuseram abordagens para representar formalmente dados sobre experimentos

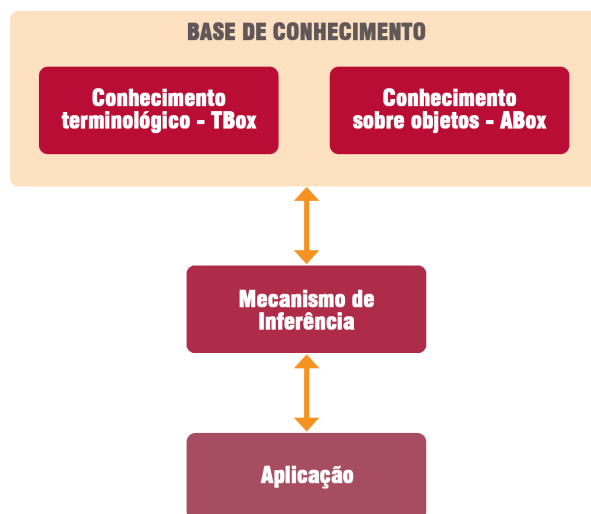


Figura 2.5: Estrutura de uma Ontologia. Autor

em ES. A revisão da literatura mostrou que a maioria dos estudos focou e representar todo o domínio do ES, ou experimentos em geral, esse fator é um agravante para elaboração dos modelos, pois, à quantidade de detalhes pode variar significativamente. Estes trabalhos estão descritos a seguir. A contribuição deles é discutida ao final desta sessão.

Durante nossa revisão de literatura encontramos: (Garcia et al., 2008) Ontologia para Experimentos Controlados em Engenharia de Software, (Scatalon et al., 2011) Empacotando Experimentos Controlados Usando uma Abordagem Evolutiva Baseada em Ontologia (S), (da Cruz et al., 2012) Uma Ontologia Fundamental para Apoiar Experimentos Científicos, (Blondet et al., 2016) ODE: uma Ontologia para Projeto Numérico de Experimentos, (Soldatova e King, 2006) Uma Ontologia de Experimentos Científicos, (Gelernter e Jha, 2016) Desafios na Avaliação de Ontologia, (Cruzes et al., 2007) Extraíndo Informações da Engenharia de Software Experimental papéis. No entanto, todos esses trabalhos não tratam experimentos de LPS.

O trabalho de Garcia et al. (Garcia et al., 2008), Poveda-Villalón et al. (Scatalon et al., 2011) e Cruz et al. (da Cruz et al., 2012) se destaca em nosso contexto por propor e modelar ontologias específicas para experimentos em engenharia de software. O trabalho de (Garcia et al., 2008) propõe, através de diagramas de classes UML, uma ontologia para experimentos controlados em engenharia de software denominada EXPERontology. Com o objetivo de ser uma ferramenta de transferência de conhecimento para auxiliares de pesquisadores e revisores, além de propor meta-análises, conduzir e avaliar experimentos controlados. O trabalho de Poveda-Villalón et al. (Scatalon et al., 2011) é uma evolução do trabalho de Garcia et al. (Garcia et al., 2008), mas focado na evolução desta ontologia

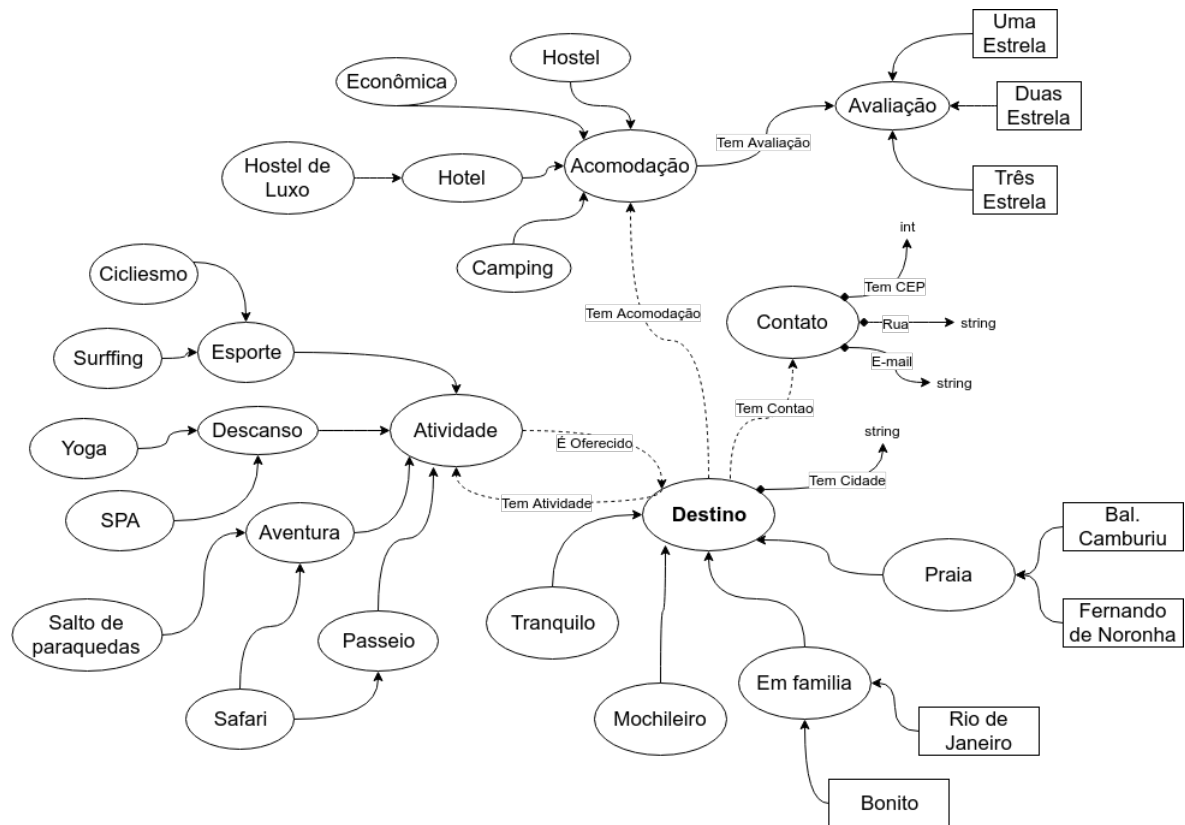


Figura 2.6: Exemplo de uma Ontologia para o domínio: Destino de Viagem. Autor

proposta. O trabalho de Cruz et al. (da Cruz et al., 2012) apresenta uma ontologia chamada OVO (Open onVence Ontology) na qual é inspirada por três teorias: (i) O ciclo de vida de experimentos científicos, (ii) Open Provent (OPM) e (iii) Unified Foundational Ontology (OVNI) Este modelo OVO pretende ser uma referência para modelos conceituais que podem ser usados por pesquisadores para explorar a semântica de metadados.

Por outro lado, os trabalhos de Blondet et al. (Blondet et al., 2016) e Soldatova e King (Soldatova e King, 2006) tratam ontologias no contexto geral de experimentos. O trabalho de Blondet et al. (Blondet et al., 2016) traz uma proposta de ontologia para DoE (Designs of Experiments) para apoiar as decisões de processo sobre o DoE. O trabalho de Soldatova e King (Soldatova e King, 2006) propõe a ontologia da EXPO que é uma mediana da ontologia SUMO (Suggested Upper Merged Ontology). Esta ontologia visa especificar os experimentos formalizando e generalizando os conceitos de design, metodologias e representação de resultados. Este trabalho é o único que usa o modelo OWL-DL para representar a ontologia.

O trabalho de Gelernter e Jha (Gelernter e Jha, 2016) dá uma visão geral sobre os desafios de avaliar uma ontologia, mas não trata da ontologia para experimentos.

Finalmente, o trabalho de Cruzes et al. (Cruzes et al., 2007) trata de uma técnica para extrair meta-informação de experimentos em engenharia de software, entendemos que este assunto está relacionado a este artigo, pois estaremos gerando através da ontologia proposta diversos metadados sobre experimentos em SPL.

Apesar de não haver um padrão para modelar o conhecimento de experimentação em ES a literatura apresenta estratégias diferentes neste sentido, como pode ser notado nos trabalhos de (Garcia et al., 2008) e (Scatalon et al., 2011). Desse modo a modelagem de ontologia de para este trabalho será aplicado para experimento em LPS, de tal modo que, será possível fazer inferência de LPS na ontologia e extrair informações. Assim tais trabalhos relacionados são fundamentais para realização deste trabalho de mestrado.

Até o momento não há trabalhos de sistema de recomendações para voltados a experimentos em ES, e nem em LPS. Outros trabalhos relacionados mais próximos já foram apresentado na Seção 2.4 e não são objetos de evolução de experimento gerais em ES.

2.7 Considerações Finais

Este capítulo apresentou os conceitos essenciais sobre a abordagem de LPS como variabilidades e variantes, apresentando o *MobileMedia* como exemplo para melhor entendimento dos conceitos, bem como o framework de engenharia de LPS proposto por /citePohl et al. (2005)?.

Em relação à experimentação em ES, foi apresentado os principais elementos como as variáveis dependentes e independentes, as atividades de um processo experimental como definição, planejamento, operação, análise e interpretação, apresentação e empacotamento. Sobre qualidade de experimentos em ES foi introduzido conceitos preliminares sobre a qualidade de experimentos como viés, validade interna e validade externa e também algumas abordagens para aplicar uma avaliação de qualidade sobre os experimentos em ES.

Quando foi tratado de ontologia, foi descrito a definição universal de ontologia e a definição voltada para computação. Nessa temos que a ontologia é uma modelagem formal de dados para representação de um determinado domínio, para isso, existe a *TBox* para representar a lógica descritiva e a *ABox* para representar os indivíduos pertencentes a ontologia. Assim como em LPS foi apresentado um exemplo (Destino de Viagem) para facilitar o entendimento dos conceitos abordados.

Por fim, temos os trabalhos relacionados traçando um paralelo de modelagem para representação do domínio de LPS eom este trabalho.

No próximo capítulo, será apresentado uma ontologia para experimento em LPS, a SMartOntology.

Uma Ontologia para Experimentos em LPS - SMartOntology

3.1 Considerações Iniciais

Este tópico apresenta conceitos fundamentais sobre a proposta de ontologia SMartyOntology. Inicialmente foi realizado um processo de concepção do modelo em seguida foi desenvolvido o projeto desenvolvido se obter modelo. Com a finalidade de avaliação foi desenvolvido um exemplo de aplicação por meio de uma predição de recomendação qual artefato de LPS pode ser usado por um dado template de experimento. Ao final desse capítulo foi realizada uma avaliação empírica do modelo, por meio de uma ferramenta que avalia pontos de falha do modelo.

3.2 Concepção

Com base na tipologia proposta por (Almeida e Bax, 2003) definimos o tipo de ontologia proposto na Tabela Tabela - 3.1:

Ou seja, quanto à função é uma ontologia de domínio, quanto ao grau de formalismo é uma ontologia semi formal, quanto à aplicação é uma ontologia de especificação, quanto à estrutura é uma ontologia de domínio e quanto ao conteúdo é tanto uma ontologia para modelagem de conhecimento quanto para aplicação e de domínio.

Foi aplicado o seguinte processo de elaboração da ontologia (Fernanda, 2007):

- Definição e estruturação dos termos por meio de classes;

Tabela 3.1: Type of the Proposed Ontology

Approach	Rating	Description
As for the Mizoguchi function; Vanwelkenbuyse; Ikeda (1995)	Domain ontologies	Reusable in the domain, about concepts and their activities and rules that
As for the degree of Uschold formalism; Gruninger (1996)	Semi-formal	Expressed in natural language and structured way
As for the Jasper application; Uschold (1999)	Ontologies as specification	An ontology is created for for documentation and development of software
As for the Haav structure; Lubi (2001)	Domain ontologies	Describe vocabulary related such as medicine or auto
As for Van-Heijst content; Schreiber; Wielinga (2002)	Knowledge modeling ontologies	Specify knowledge concepts semantically rich internal for use in the domain of
	Application ontologies	Contains the definitions in an application
	Domain ontologies	Express concepts that are given domain of knowledge

- Estabelecimento de propriedades (atributos) inerentes ao conceito representado por um termo;
- Povoamento da estrutura que satisfaçam um conceito e as suas propriedades;
- Estabelecimento de relações entre os conceitos;
- Elaboração de sentenças para restringir inferências de conhecimento baseadas na estrutura.

Inicialmente foi desenvolvido um grafo para o modelo inicial da ontologia SMartyOntology, principalmente para validar os termos por meio de classes, subclasses e os relacionamento entre elas.

A Figura - 3.1 representa este modelo inicial contendo a definição principal das classe de mais alto nível para o experimento, SPL Experiment, Documentation, Template, Evaluation, Discussion, Analysis, Execution e Planning. Em seguida evoluímos para possíveis subclasses delas, onde definimos as sub-classes em Tabela - 3.2:

A criação deste modelo inicial de ontologia se baseada no trabalho de mapeamento sistemático de experimentos em SPL (Furtado, 2018), por meio de uma análise exploratória dos dados levantados nesse mapeamento. O mapeamento servir como guia de informações e metadados de experimentos.

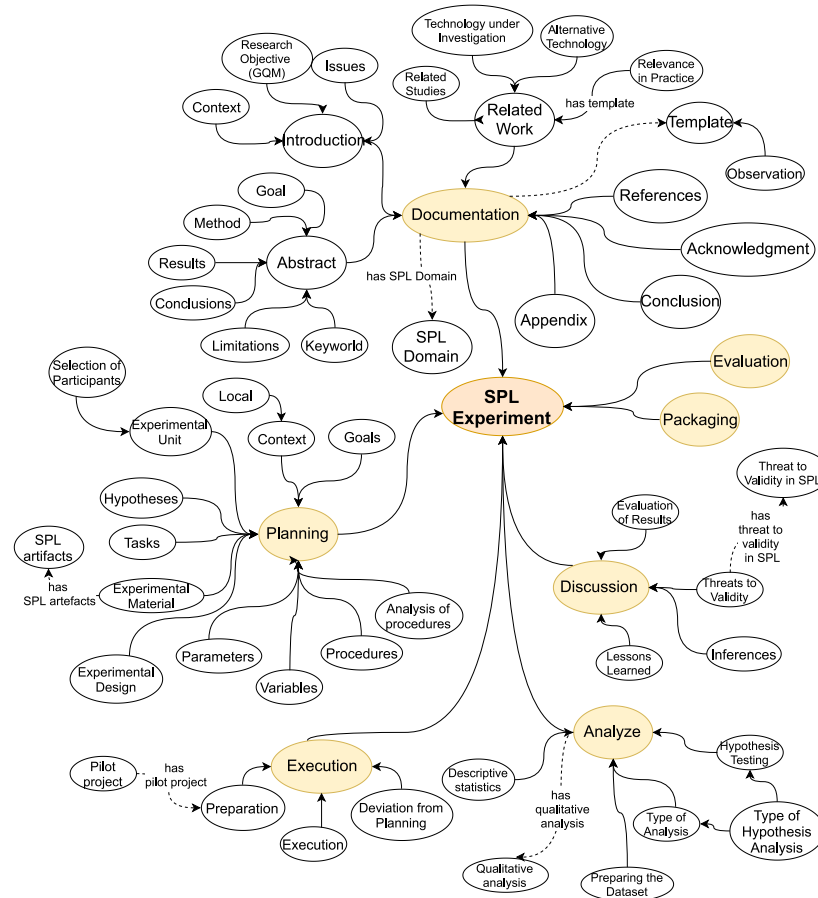


Figura 3.1: Initial graph of ontology.

O mapeamento sistemático se baseou no temple experimental de Wohlin, que traz 5 pilares para elaboração de ESE, são eles: Definição, Planejamento, Operação, Análise e Interpretação (Wohlin et al., 2012b).

A Figura - 3.2, apresenta uma modificação do modelo conceitual original, aplicando uma clusterização baseada nos pilares do Wohlin. Esta clusterização foi necessária para compreensão mais abstrata das relações entre os termos do domínio levantados no modelo conceitual original. Dessa forma foi possível validar as classes pai do grafo inicialmente proposto.

Em seguida, um diagrama de classes foi criado para uma representação mais formal da modelagem inicial. Nessa representação, a relação entre os termos (classes) e suas propriedades (atributos) ficou mais clara, o modelo TBox. Essa forma de representação destacou a relação principal quando definimos a composição da classe Experiment e ExperimentSPL em quase todas as outras subclasses. Nesta representação também foi explícita a tipificação das propriedades, modelo ABox.

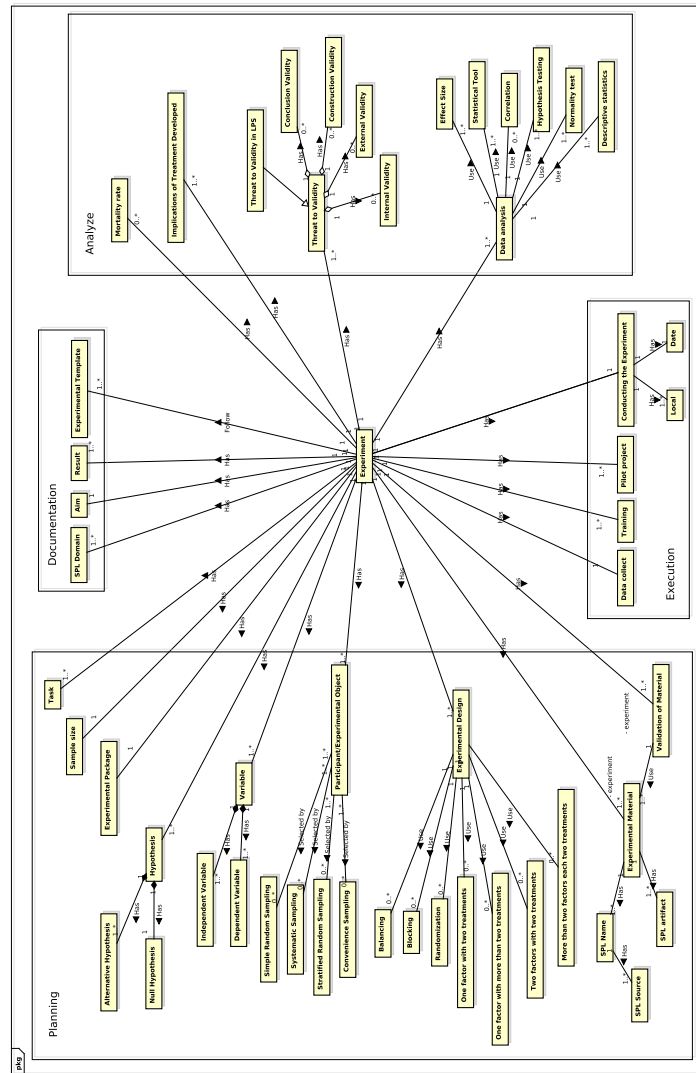


Figura 3.2: Conceptual Model Clustering.

A Figura - 3.3 apresenta diagrama de classe gerado para SMartyOntology.

Posteriormente, definimos usar a ferramenta Protégé para a construção oficial da ontologia. Esta ferramenta dispõe de uma interface gráfica para edição de ontologias e uma arquitetura para a criação de ferramentas baseadas em conhecimento. Pode ser usada tanto por desenvolvedores de sistema como por especialistas em domínio para criar bases de conhecimento, permitindo representar facilmente o conhecimento de uma área. Este editor é capaz de tratar classes, com sua definição e exemplos, simultaneamente (Musen, 2015).

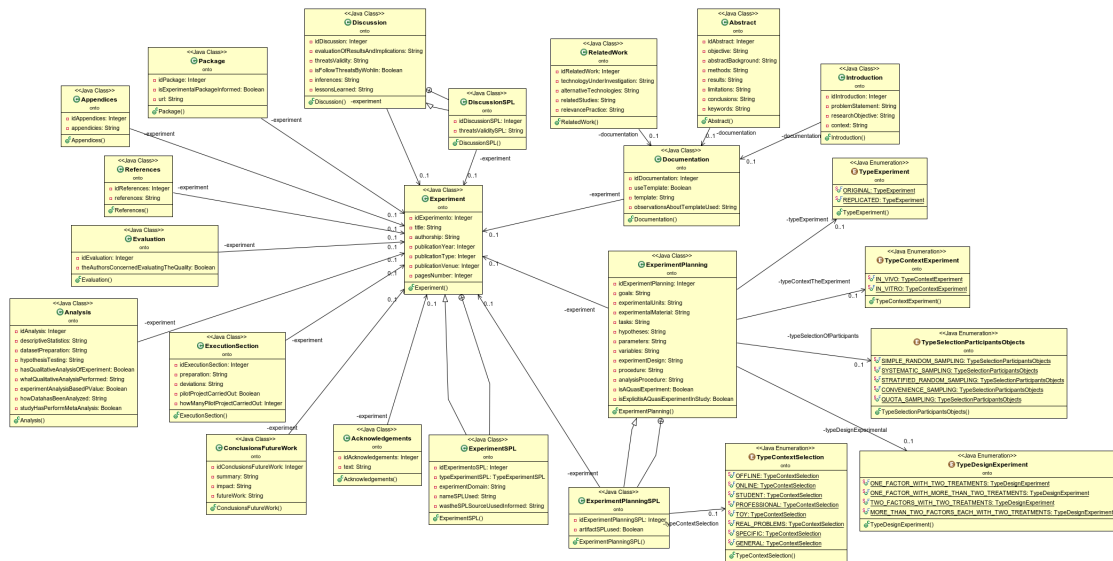


Figura 3.3: Class Diagram for proposal of the ontology model.

3.3 Projeto

Definimos a criação da ontologia usando OWL construindo as classes e subclasses para representar os elementos da ontologia.

3.3.1 Modelagem com Protégé

O padrão OWL foi usado no SMartyOntology para definir todos os elementos, classes e subclasses.

Fazendo uma analogia ao diagrama de classe, no Protégé, classes, os atributos de classes e seus relacionamentos estão em um contexto de entidades. As classes e hierarquias são definidos na aba Class, os relacionamentos são definidos na aba Propriedades de Objetos e os atributos são definidos na aba Propriedade dos Dados.

Definimos nossas entidades com base do diagrama de classe construído na fase de concepção, com a seguinte estrutura (i) definição de classes (ii) definição das propriedades de objetos, (iii) definição das propriedades dos dados.

Definição de classes

No Protégé, definimos uma classe raiz chamada Thing na qual todas as outras classes são filhas dela. A Figura - 3.4 apresenta a definição da classe Experiment dentro do modelo, neste momento a definição é composta pelo nome da classe e seus principais relacionamentos (Equivalent To e SubClass Of).

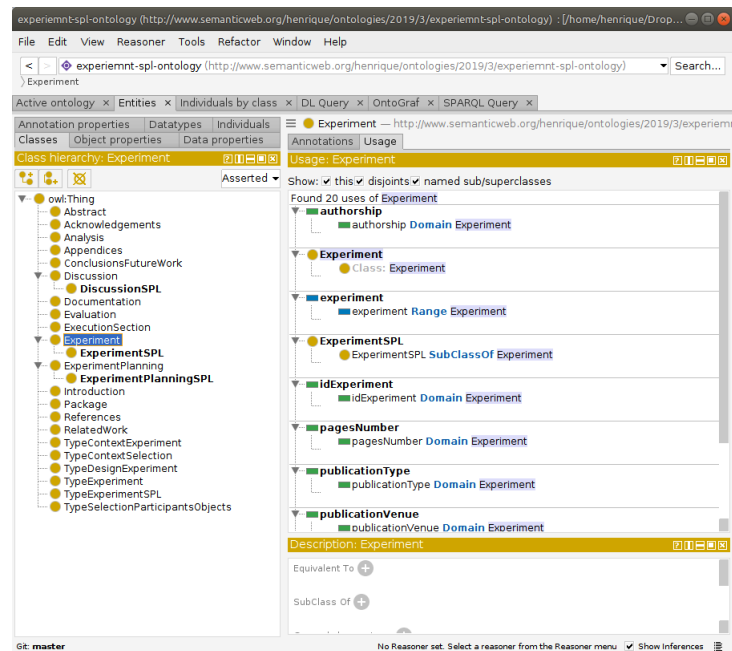


Figura 3.4: Definição da classe Experiment no Protégé.

Definição das propriedades de objetos

No Protégé, definimos uma propriedade de objeto raiz chamada `topObjectProperty` na qual todas as outras propriedades são filhas dela. A Figura - 3.5 apresenta a definição de propriedade de objeto para chamada `?documentation?` dentro do modelo, neste momento a definição é composta pelo nome da propriedade e seus relacionamentos com classes, Domains e Ranges.

Definição das propriedades dos dados

No Protégé, definimos uma propriedade de dados raiz chamada `topDataProperty` na qual todas as outras propriedades são filhas dela. Para cada Propriedade de Dado definimos um conjunto de classes de seu Domínio e um conjunto de tipos de dados (predefinido) de seu Alcance, por exemplo, a propriedade de dado `?nameSPLUsed?` possui como classe de domínio `ExperimentSPL` e como alcance um `xsd:string`. A Figura - 3.6 apresenta a definição de propriedade de dado para chamada `?nameSPLUsed?` dentro do modelo, neste momento a definição é composta pelo nome da propriedade e seus relacionamentos, Domains (uma classe) e Ranges (uma tipagem).

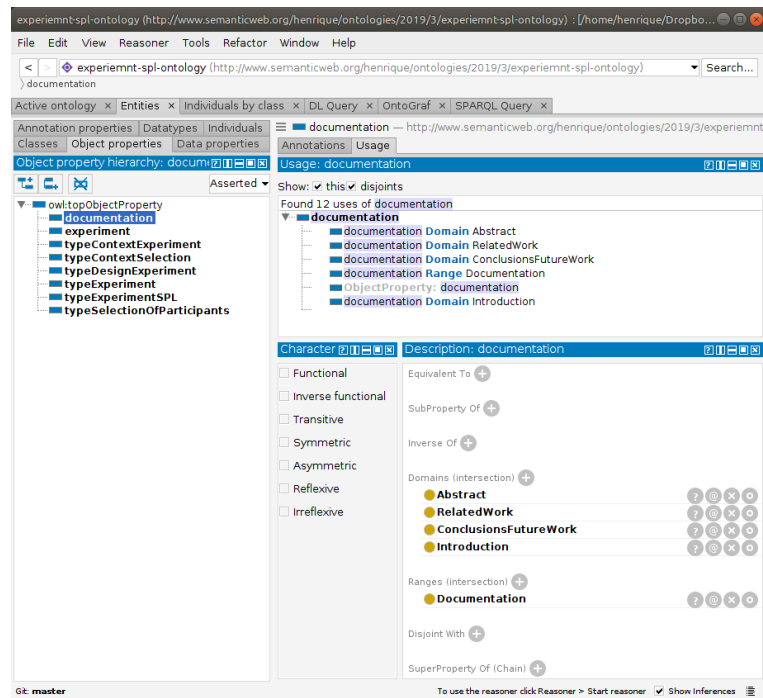


Figura 3.5: Definição da propriedade de objeto documentation no Protégé.

Artefato gerado pelo Protégé

Ao final, o Protégé gera um arquivo .owl contendo a definição do modelo. A Figura 3.7 fornece uma visão geral do formato do grafo do projeto, contendo todas as classes e suas propriedades de objeto e propriedade de dados. Usamos a ferramenta WebVOWL (Lohmann et al., 2016) para gerar essa visão.

3.3.2 Povoamento com Python

A próxima etapa, após a modelagem da ontologia, foi inserir os indivíduos na mesma, ou seja, popular a ontologia para realização de inferências futuras. Apesar do Protégé ter capacidade para realizar essa operação, optamos por utilizar um script para realizar tal tarefa, visto que, no Protégé o processo de inserir indivíduos na ontologia é realizado de modo manual, por meio do menu ?Individuals by Class?, onde é preciso selecionar propriedade por propriedade para cada indivíduo. Sabendo que para cada indivíduo temos 84 propriedades de dados mais 8 propriedade de objetos, somando um total de 92 relacionamentos para cada indivíduo. Portanto, sabendo que temos 174 indivíduos a serem inseridos em uma carga inicial, seriam mais de 16000 iterações manuais no Protégé.

Dado essa estimativa de operações manuais optamos pela utilização de uma ferramenta script a fim de facilitar e automatizar a inserção dos indivíduos e posteriormente a inserção

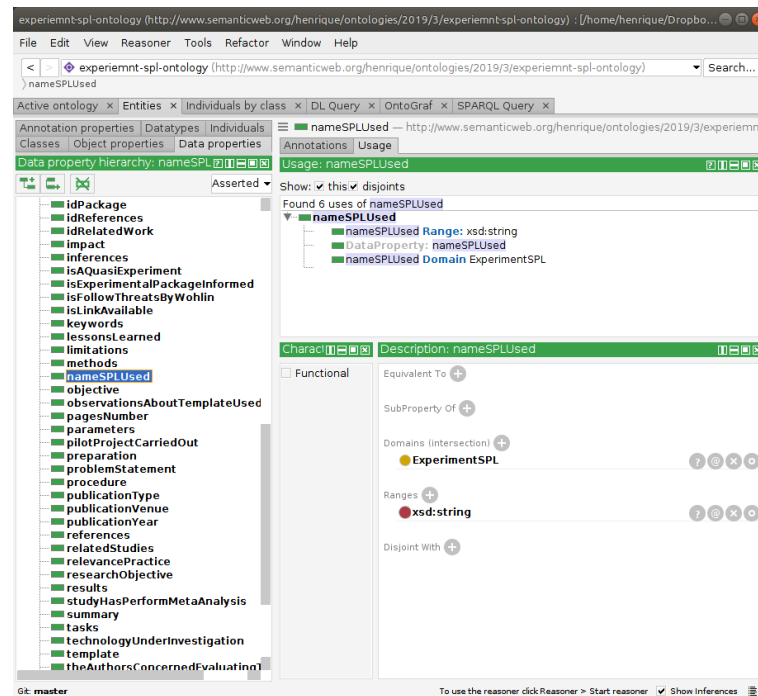


Figura 3.6: Definição da propriedade de dado nameSPLUsed no Protégé.

de novos indivíduos. Escolhemos a linguagem de programação Python por fornecer bibliotecas prontas, tanto para manipulação de dados em planilhas (Pandas), como para ontologia em arquivos .owl (OwlReady2).

Script

Para criação do script utilizamos outra ferramenta do Python chamada Jupyter Notebook (Pérez e Granger, 2007) para auxiliar na execução e validação de código. O processo do script se assemelha a um processo de ETL (Extract Transform Load), onde extraímos os dados originais da planilha, desenvolvida no trabalho de revisão sistemática sobre experimentos em LPS, e manipulamos estes dados para inserir na ontologia, seguindo a modelagem inicial construída no Protégé.

Para leitura dos dados da planilha usamos a biblioteca Pandas (McKinney, 2010), ela nos retorna um objeto do tipo Dataframe que chamamos de `df`, no qual é a representação da própria planilha no ambiente de programação Python.

Os dados da planilha estão estruturados na seguinte forma, cada linha representa um artigo encontrado no mapeamento sistemático, e cada coluna representa um dado extraído deste artigo. Segue o mapeamento de cada dado.

Com relação ao Experimento temos: ID, Title, Authorship, Publication year, Publication type, Publication venue, Pages number, dados de Experimentos em SPL temos: Is it Real or Academic SPL?, SPL Name used, Was the SPL source used informed? (Y/N) (If yes, which one?).

Com relação a Documentation temos: Does it use template? (Y/N)?, If yes, what template?, Observations about the template used. Para o Abstract temos: Objective (What is the question addressed with this research?), Abstract - Background (Why is this research important?), Methods (What is the statistical context and methods applied?), Results (What are the main findings? Practical implications?), Limitations (What are the weakness of this research?), Conclusions (What is the conclusion?) e Keywords. Para o Introduction Section temos: Problem statement (What is the problem? Where does it occur? Who has observed it? Why is it important to be solved?), Research objective (GQM) (What is the research question to be answered by this study?), Context (What information is necessary to understand whether the research relates to a specific situation

(environment)?). Para o Related Work Section temos: Technology under investigation (What is necessary for a reader to know about the technology to reproduce its application?), Alternative technologies (How does this research relate to alternative technologies? What is the control treatment?), Related studies (How this research relates to existing research (studies)? What were the results from these studies?), Relevance to practice (How does it relate to state of the practice?). Para o Conclusions and Future Work Section temos: Summary (The purpose of this section is to provide a concise summary of the research and its results as presented in the former sections), Impact (Description of impacts with regard to cost, schedule, and quality, circumstances under which the approach presumably will not yield the expected benefit), Future Work (What other experiments could be run to further investigate the results yielded or evolve the Body of Knowledge).

Com relação ao Experiment Planning temos: Goals (Formalization of goals, refine the important constructs of the experiment's goal), Experimental Units (From which population will the sample be drawn? How will the groups be formed (assignment to treatments)? Any kind of randomization and blinding has to be described?), Experimental Material (Which objects are selected and why?), Tasks (Which tasks have to be performed by the subjects?), Hypotheses (What are the constructs and their operationalization? They have to be traceable derived from the research question respectively the goal of the experiment), Parameters (What are the constructs and their operationalization? They have to be traceable derived from the research question respectively the goal of the experiment), Variables (What are the constructs and their operationalization? They have to be traceable derived from the research question respectively the goal of the experiment), Experiment Design (What type of experimental design has been chosen?), Procedure (How will the experiment (i.e., data collection) be performed? What instruments, materials, tools will be used and how?) Analysis Procedure (How will the data be analyzed?), Is it a quasi-experiment? (Y/N) If yes, is it explicit in the study?, Is it an Original or Replicated Experiment?, How was the selection of participants/experimental objects? - Simple random sampling; - Systematic sampling; - Stratified random sampling; - Convenience sampling; - Quota sampling, Context of the experiment (in vivo, in vitro, ...), Design Experimental: - One factor with two treatments; - One factor with more than two treatments; - Two factors with two treatments; - More than two factors each with two treatments, SPL artifact used, Context Selection (Off-line vs. on-line, Student vs. professional, Toy vs. real problems, Specific vs. general).

Com relação a Execution (Operation) temos: Preparation (What has been done to prepare the execution of the experiment (i.e., schedule, training), Deviations from the

Plan (Describe any deviations from the plan, e.g., how was the data collection actually performed?), The pilot project was carried out? (Y/N) If Yes, how many?

Com relação a Analysis (Analysis and Interpretation) temos: Descriptive statistics (What are the results from descriptive statistics?), Data set preparation (What was done to prepare the data set, why, and how?), Hypothesis testing (How was the data evaluated and was the analysis model validated?), Do it have qualitative analysis of the experiment? (Y/N), If yes, what qualitative analysis was performed?, How the data has been analyzed? (Ex: Correlation, Hypothesis Test, meta-analysis), Is the conclusion of the experiment analysis based on P-value? (Y/N), Did the study perform meta-analysis? (Y/N).

Com relação a Discussion temos: Evaluation of Results and Implications (Explain the results and the relation of the results to earlier research, especially those mentioned in the Background section), Threats to Validity (How is validity of the experimental results assured? How was the data actually validated?) (Follow are the 4 threats proposed by Wohlin: internal, external, construct and conclusion? (Y/N)), Inferences (Inferences drawn from the data to more general conditions), Lessons learned (Which experience was collected during the course of the experiment), Threats to validity in SPL.

Com relação a Evaluation temos: The authors were concerned with evaluating the quality of the experiments? (Y/N)

Com relação a Package temos: Is the experimental package informed? (Y/N) (If yes, what URL? And the link is still available? (Y/N))

Outros dados: Acknowledgements Section (Sponsors, participants, and contributors who do not fullfit the requirements for authorship should be mentioned), References Section (All cited literature has to be presented in the format requested by the publisher, Appendices Section (Experimental materials, raw data, and detailed analyses, which might be helpful for others to build upon the reported work should be provided)

Manipulação dos Dados

Para inserir os indivíduos na SMartyOntology, foi preciso manipular os dados da planilha com as seguintes operações.

Primeira operação, split de dados de uma única coluna para duas propriedades de dados da ontologia. Este caso ocorreu para os seguintes dados, as propriedades e prefixadas com `?_?` foram tratadas no próximo passo:

- `?Was the SPL source used informed?` (Y/N) (If yes, which one?)? para `_informedSPL` e `sourceSPL`;

- ?The pilot project was carried out? (Y/N) If Yes, how many?? para `_hasPilot` e `_howManyPilot`;
- ?Is it a quasi-experiment? (Y/N) If yes, is it explicit in the study?? para `_hasQuasiExperiment` e `quasiExperiment`;
- ?'Threats to Validity (How is validity of the experimental results assured? 'How was the data actually validated?) (Follow are the 4 threats proposed by Wohlin: internal, external, construct and conclusion? (Y/N)))? para `_hasThreatsValidityByWolin` e `threatsValidity`.

Segunda operação, transformação de dados booleanos, strings vazias e números. Foi desenvolvido tres funções de conversão, (i) `convertToBoolean`, (ii) `convertStringEmpty` e (iii) `convertToNumber`. Este caso ocorreu para os seguintes dados:

- ?Does it use template? (Y/N)?? para `useTemplate` aplicando o método `convertToBoolean`;
- ?If yes, what template?? para `template` aplicando o método `convertStringEmpty`;
- ?Do it have qualitative analysis of the experiment? (Y/N)? para `hasQualitativeAnalysis` aplicando o método `convertToBoolean`;
- ?Did the study perform meta-analysis? (Y/N)? para `hasMetaAnalysis` aplicando o método `convertToBoolean`;
- `_informedSPL` para `informedSPL` aplicando o método `convertToBoolean`;
- `_hasQuasiExperiment` para `hasAQuasiExperiment` aplicando o método `convertToBoolean`;
- `_hasPilot` para `hasPilot` aplicando o método `convertToBoolean`;
- `_howManyPilot` para `howManyPilot` aplicando o método `convertToNumber`;
- `_hasThreatsValidityByWolin` para `hasThreatsValidityByWolin` aplicando o método `convertToBoolean`;
- `_hasPackage` para `hasExperimentalPackage` aplicando o método `convertToBoolean`;

Terceira operação, separação do conjunto de dados com informação explícita de SPL. Separamos o conjunto total de 174 artigos em artigos de experimentos que contêm informações explícitas de SPL (152) e artigos que não contêm informação explícitas de SPL (22), para criação dos indivíduos conforme as classes modeladas na ontologia com atributos pertinentes a SPL.

Quarta operação, padronização de dados constantes e faltantes. Foi preciso padronizar os dados em casos faltantes foi atribuído um padrão. Isso ocorreu para os seguintes dados:

- "Is it Real or Academic SPL?" foi definido o seguinte conjunto de dados [REAL, ACADEMY] sendo o padrão "ACADEMY";
- "Is it an Original or Replicated Experiment?" foi definido o seguinte conjunto de dados [ORIGINAL, REPLICATED] sendo o padrão "REPLICATED"
- "How was the selection of participants/experimental objects? - Simple random sampling; - Systematic sampling; - Stratified random sampling; - Convenience sampling; - Quota sampling." foi definido o seguinte conjunto de dados [SIMPLE_RANDOM_SAMPLING, SYSTEMATIC_SAMPLING, STRATIFIED_RANDOM_SAMPLING, CONVENIENCE_SAMPLING, QUOTA_SAMPLING] sendo o padrão "CONVENIENCE_SAMPLING";
- "Context of the experiment (in vivo, in vitro, ...)" foi definido o seguinte conjunto de dados [IN_VIVO, IN_VITRO] sendo o padrão "IN_VITRO";
- "Design Experimental: - One factor with two treatments; - One factor with more than two treatments; - Two factors with two treatments; - More than two factors each with two treatments." foi definido o seguinte conjunto de dados [ONE_FACTOR_WITH_TWO_TREATMENTS, ONE_FACTOR_WITH_MORE_THAN_TWO_TREATMENTS, TWO_FACTORS_WITH_TWO_TREATMENTS, MORE_THAN_TWO_FACTORS_EACH_WITH_TWO_TREATMENTS] sendo o padrão "ONE_FACTOR_WITH_TWO_TREATMENTS";
- "Context Selection (Off-line vs. on-line, Student vs. professional, Toy vs. real problems, Specific vs. general)" foi definido o seguinte conjunto de dados [OFFLINE, ONLINE, STUDENT, PROFESSIONAL, TOY, REAL_PROBLEMS, SPECIFIC, GENERAL] sendo o padrão "GENERAL";

Inserção do indivíduos e criação dos relacionamentos

Inicialmente fazemos a leitura do modelo da ontologia gerada pelo Protégé (arquivo .owl) por meio da biblioteca Owlready2, com isso temos um objeto chamado ?onto? onde a

partir dele podemos executar operações de inserção de indivíduos e outras operações na ontologia no ambiente de programação Python.

O processo de inserção dos dados extraídos da planilha se dá por, percorrer linha a linha da planilha, e criar os indivíduos um a um de cada classe modelada na ontologia com seus respectivos atributos. Para isso foi necessário criar vários métodos a fim de facilitar a compreensão do script.

Foi criado dois laços para percorre os dois subconjuntos de dados, um com informações explícitas de SPL chamado de `df_spl` e outro chamado `df_exp`. O primeiro laço sobre `df_spl` invoca a seguinte sequencia de métodos: `registreExperimentSPL` \wedge `registreExperimentPlanningSPL` \wedge `registreDiscussionSPL` \wedge `registerCommons`. O segundo laço sobre `df_exp` invoca a seguinte sequencia de métodos: `registreExperiment` \wedge `registreExperimentPlanning` \wedge `registreDiscussion` \wedge `registerCommons`.

Listing 3.1: The first loop on `df_spl`

```
for idx, row in df_spl.iterrows():
    experimentSPL = registreExperimentSPL(idx, row)

    experimentPlanningSPL = registreExperimentPlanningSPL(idx, row)
    experimentPlanningSPL.experiment.append(experimentSPL)

    discussion = registreDiscussionSPL(idx, row)
    discussion.experiment.append(experimentSPL)

    registerCommons(experimentSPL, idx, row)
```

Listing 3.2: The second loop on `df_exp`

```
for idx, row in df_exp.iterrows():
    experiment = registreExperiment(idx, row)

    experimentPlanning = registreExperimentPlanning(idx, row)
    experimentPlanning.experiment.append(experiment)

    discussion = registreDiscussion(idx, row)
    discussion.experiment.append(experiment)

    registerCommons(experiment, idx, row)
```


O método `registreExperimentSPL` executa o método `registreExperimentCommons` e retorna uma instância de indivíduo da ontologia da classe `onto.ExperimentSPL`.

O método `registreExperiment` executa o método `registreExperimentCommons` e retorna uma instância de indivíduo da ontologia da classe `onto.Experiment`.

O método `registreExperimentCommons` recebe uma instância tanto de `onto.ExperimentSPL` ou `onto.Experiment` e atribui as variáveis comuns para ambas as classes.

O método `registreExperimentPlanningSPL` executa o método `registreExperimentPlanningCommons` e retorna uma instância de indivíduo da ontologia da classe `onto.ExperimentPlanningSPL`.

O método `registreExperimentPlanning` executa o método `registreExperimentPlanningCommons` e retorna uma instância de indivíduo da ontologia da classe `onto.ExperimentPlanning`.

O método `registreExperimentPlanningCommons` recebe uma instância tanto de `onto.ExperimentSPL` ou `onto.ExperimentPlanning` e atribui as variáveis comuns para ambas as classes.

O método `registreDiscussionSPL` executa o método `registreDiscussionCommons` e retorna uma instância de indivíduo da ontologia da classe `onto.DiscussionSPL`.

O método `registreDiscussion` executa o método `registreDiscussionCommons` e retorna uma instância de indivíduo da ontologia da classe `onto.Discussion`.

O método `registreDiscussionCommons` recebe uma instância tanto de `onto.DiscussionSPL` ou `onto.Discussion` e atribui as variáveis comuns para ambas as classes.

O método `registerCommons` que recebe uma instância tanto de `onto.ExperimentSPL` ou `onto.Experiment` e executa a sequência de métodos: `registreDocumentation` - que retorna uma instância de `onto.Documentation` e atribui a instância de `experiment` a ela ; `registreAbstract` - que retorna uma instância de `onto.Abstract` e atribui a instância de `documentation` a ela ; `registreIntroduction` - que retorna uma instância de `onto.Introduction` e atribui a instância de `documentation` a ela ; `registreRelatedWork` - que retorna uma instância de `onto.RelatedWork` e atribui a instância de `documentation` a ela ; `registreConclusionsFutureWork` - que retorna uma instância de `onto.ConclusionsFutureWork` e atribui a instância de `documentation` a ela ; `registreExecutionSection` - que retorna uma instância de `onto.ExecutionSection` e atribui a instância de `experiment` a ela ; `registreAnalysis` - que retorna uma instância de `onto.Analysis` e atribui a instância de `experiment` a ela ; `registreAcknowledgements` - que retorna uma instância de `onto.Acknowledgements` e atribui a instância de `experiment` a ela ; `registreReferences` - que retorna uma instância de `onto.References` e atribui a instância de `experiment` a ela ; `registreAppendices` - que retorna uma instância de `onto.Appendices` e atribui a instância de `experiment` a ela ; `registreEvaluation` - que retorna uma instância de `onto.Evaluation` e atribui a instância de `experiment` a ela ; `registrePackage` - que retorna uma instância de `onto.Package` e atribui a instância de `experiment` a ela.

Artefato e validação

Ao final dos dois laços temos o objeto `?onto?` populado com os indivíduos, e por meio da biblioteca `Owlready2` geramos um novo arquivo `.owl` contendo a modelagem da ontologia mais a população de indivíduos. O arquivo da modelagem inicial contém 66,3 kB, e o arquivo populado contém 4,8 MB.

No final do script tem existe um passo de validação onde conferimos a quantidade de linhas da planilha com a quantidade de indivíduos inserido na ontologia, pode ser visto em 3.3.

Listing 3.3: Stap validation

```
assert len (onto.Experiment.instances ()) == df.shape [0]
```

3.4 Exemplo de Aplicação

Foi criado também um exemplo de recomendação com a seguinte questão de recomendação, `?Qual artefato de SPL usar dos experimentos que usam a mesma template experimental mais recorrente?`. Com base nesta questão foi criado um script Python utilizando as bibliotecas `Pandas` e `OwlReady2`, para responder a mesma.

O script começa carregando a ontologia preenchida e, em seguida, executamos a consulta `SPARQL ??` para retornar os agrupados pela template experimental, estes indivíduos são do tipo `Documentation`. Na sequência, filtramos da lista inicial de indivíduos que usam modelos, aqueles que usam o modelo mais recorrente. Com esses indivíduos em mãos, buscamos seu relacionamento com indivíduos do tipo `ExperimentPlanningSPL` para extrair informações de artefatos de SPL. Assim, podemos recomendar qual artefato de SPL, com base no artefato mais usado desses indivíduos filtrados.

O script começa carregando a ontologia povoada, e em seguida extrai os indivíduos que usam template experimental, estes indivíduos são do tipo `Documentation`. Atribuímos este resultado a um objeto do tipo `Pandas` para executar a função `.values_counts()` que retorna o agrupamento somado das templates mais usadas. Na sequência, filtramos da lista inicial dos indivíduos que usam templates, aqueles que usam a template mais recorrente. Com estes indivíduos em mãos, buscamos o relacionamento deles com os indivíduos do tipo `ExperimentPlanningSPL` para extrair a informação dos artefatos de SPL. Então podemos recomendar qual artefato de SPL, com base no artefato mais usado destes indivíduos filtrados.

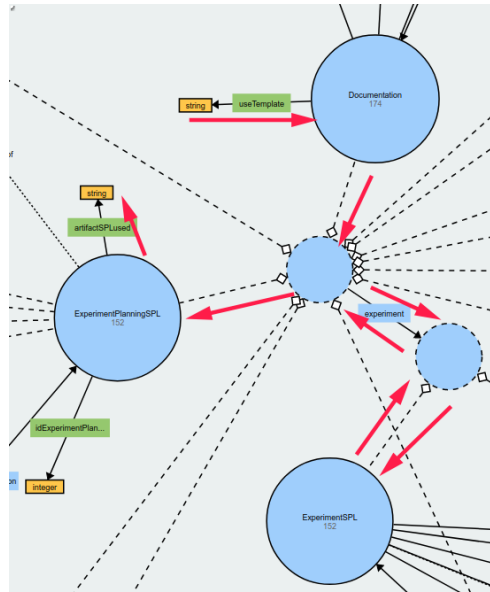


Figura 3.8: Graph Path to Recommendation Example.

O grafo da Figura - 3.8 mostra o caminho percorrido entre as classes e propriedades de objetos e dados para chegar no artefato de SPL a ser recomendado.

O exemplo de recomendação apresenta como podemos criar mecanismos de inferência no modelo de ontologia proposto neste trabalho. Foi possível avaliar que com algumas consultas e verificações é possível extrair informações e meta informações sobre experimentos em SPL.

Este exemplo percorre três classes (Documentation, ExperimentSPL e ExperimentPlanningSPL) de 24 classes, duas propriedades de objetos (documentation e experimet) de 8 propriedades e três propriedades de dados (useTemplate, template e artifactSPLused) de 87 propriedades de dados do modelo proposto. O cálculo . estima que neste exemplo estamos usando 0,1

O exemplo apresenta como podemos criar mecanismos de inferência no modelo de ontologia proposto neste trabalho. Foi possível avaliar que com algumas consultas e verificações é possível extrair informações e meta informações sobre experimentos em SPL.

Este exemplo percorre três classes (Documentation, ExperimentSPL e ExperimentPlanningSPL) de 24 classes, duas propriedades de objetos (documentation e experimet) de 8 propriedades e três propriedades de dados (useTemplate, template e artifactSPLused) de 87 propriedades de dados do modelo proposto. O cálculo . estima que neste exemplo estamos usando 0,1% da capacidade de resposta que o modelo possa responder.

No caso do exemplo de recomendação a primeira parte do retorna a template mais recorrente. O resultado é a saída do script: ?Template more frequency is Wohlin? seguida da Fig. que representa os dados da Tabela.

E segunda parte encontra qual artefato de SPL dos indivíduos que usam a template encontrada na primeira etapa. o resultado é a saída do script: ?Artifact more used is the "test cases"to template: Wohlin? seguido da Fig.

3.5 Avaliação Empírica

Para avaliação da ontologia seguimos duas estratégias, uma para avaliar possíveis armadilhas e falhas no modelo de ontologia proposto.

Usamos a ferramenta OOPS! para gerar avaliação do modelo de ontologia proposto. A ferramenta ajuda a detectar algumas das armadilhas mais comuns que aparecem ao desenvolver ontologias [referência].

- P01. Criando elementos polissêmicos;
- P02. Criando sinônimos como classes;
- P03. Criando o relacionamento “ is ” em vez de usar “ rdfs: subClassOf ”; “ rdf: type ” ou “ owl: sameAs ”;
- P04. Criando elementos de ontologia não conectados;
- P05. Definindo relações inversas erradas;
- P06. Incluindo ciclos em uma hierarquia de classes;
- P07. Mesclar diferentes conceitos na mesma classe;
- P08. Anotações ausentes;
- P09. Informações de domínio ausentes;
- P10. Desarticulação em falta;
- P11. Domínio ou intervalo ausente nas propriedades;
- P12. Propriedades equivalentes não explicitamente declaradas;
- P13. Relações inversas não explicitamente declaradas;

- P14. Uso indevido de “ owl: allValuesFrom ”;
- P15. Usando “ alguns não ” no lugar de “ não alguns ”;
- P16. Usando uma classe primitiva no lugar de uma definida;
- P17. Superespecialização de uma hierarquia;
- P18. Superespecialização do domínio ou intervalo;
- P19. Definir vários domínios ou intervalos nas propriedades;
- P20. Uso indevido de anotações de ontologia;
- P21. Usando uma classe diversa;
- P22. Usando diferentes convenções de nomenclatura na ontologia;
- P23. Duplicando um tipo de dados já fornecido pela linguagem de implementação;
- P24. Usando definições recursivas;
- P25. Definir um relacionamento como inverso de si mesmo;
- P26. Definindo relações inversas para uma simétrica;
- P27. Definir propriedades equivalentes erradas;
- P28. Definindo relacionamentos simétricos errados;
- P29. Definindo relacionamentos transitivos errados;
- P30. Classes equivalentes não explicitamente declaradas;
- P31. Definir classes equivalentes erradas;
- P32. Várias aulas com o mesmo rótulo;
- P33. Criando uma cadeia de propriedades com apenas uma propriedade;
- P34. Classe sem tipografia;
- P35. Propriedade não tipificada;
- P36. URI contém extensão de arquivo;
- P37. Ontologia não disponível na Web;

- P38. Nenhuma declaração de ontologia OWL;
- P39. Namespace ambíguo;
- P40. Sequestro de namespace;
- P41. Nenhuma licença declarada.

Apesar da ferramenta OOPS! ter 41 pontos de avaliação ela executa apenas 34 pontos semi-automaticamente, pois os outros dependem de domínio específico da ontologia e eles encorajam os usuários a melhorarem a ferramenta. O resultado dado pela ferramenta sugere como os elementos da ontologia poderiam ser modificados para melhorar a qualidade da ontologia. No entanto, nem todas as armadilhas identificadas devem ser interpretadas como erro,, mas como sugestões que devem ser revisadas manualmente em alguns casos.

Esta avaliação pode ajudar a descobrir erros que foram escondidos por causa da falta de informação. Por exemplo, algumas armadilhas são detectadas pela comparação de domínios e intervalos em propriedades, se não estiverem definidas, as armadilhas não podem ser identificadas. Nesse sentido, corrigindo a armadilha "Falta do domínio ou intervalo em propriedades" faz com que a ferramenta para encontrar outras armadilhas, por exemplo, "Definir relações simétricas que não possuem o mesmo domínio e alcance?".

A ferramenta elenca cada armadilha como:

- **Critical:** Corrigir a armadilha é crucial. Caso contrário, isso poderia afetar a consistência, raciocínio, aplicabilidade, etc. da ontologia;
- **Importante:** Embora não seja crítico para a função de ontologia, é importante corrigir este tipo de armadilha;
- **Minor:** Não é realmente um problema, mas ao consertá-lo, tornaremos a ontologia mais agradável.

A Tabela. apresenta o resumo do resultado ao rodar nosso modelo de ontologia proposto na ferramenta OOPS!.

No caso P08, essa armadilha consiste em criar um elemento de ontologia e não fornecer anotações legíveis a ele. Consequentemente, os elementos de ontologia não possuem propriedades de anotação que os identificam (por exemplo, `rdfs: label`, `lemon: LexicalEntry`, `skos: prefLabel` ou `skos: altLabel`) ou que os definem (por exemplo, `rdfs: comment` ou `dc: description`). Essa armadilha está relacionada às diretrizes fornecidas em [referencia 5 do OOPS].

No caso P10, a ontologia não possui axiomas desarticulados entre classes ou entre propriedades que devem ser definidas como disjuntas. Esta armadilha está relacionada com as orientações fornecidas em [6], [2] e [7].

No caso P12, a ontologia carece de informações sobre propriedades equivalentes (`owl: equivalentProperty`) nos casos de relacionamentos e / ou atributos duplicados. Os seguintes atributos podem ser definidos como equivalentes: `wasTheSPLSourceUsedInformed` e `wastheSPLSourceUsedInformed`.

No caso P13, essa armadilha aparece quando qualquer relacionamento (exceto aqueles que são definidos como propriedades simétricas usando `owl: SymmetricProperty`) não tem uma relação inversa (`owl: inverseOf`) definida dentro da ontologia. Esta armadilha aparece nos seguintes elementos: `typeSelectionOfParticipants`, `typeExperimentSPL`, `typeExperiment`, `typeDesignExperiment`, `typeContextSelection`, `typeContextExperiment`, `experiment`, `documentation`

No caso P19, o domínio ou intervalo (ou ambos) de uma propriedade (relacionamentos e atributos) é definido informando mais de uma instrução `rdfs: domain` ou `rdfs: range`. Em OWL vários `rdfs: domain` ou `rdfs: range` axiomas de alcance são permitidos, mas eles são interpretados como conjunção, sendo, portanto, equivalentes ao construto `owl: intersectionOf`. Essa armadilha está relacionada ao erro comum que aparece ao definir domínios e intervalos descritos em [7]. Esta armadilha aparece nos seguintes elementos: `documentation`, `experiment`, `typeContextExperiment`, `typeDesignExperiment`, `typeExperiment`, `typeSelectionOfParticipants`

No caso P41, os metadados de ontologia omitem informações sobre a licença que se aplica à ontologia.

* Armadilha que se aplica à ontologia em geral, em vez de elementos específicos.

Dada a análise da ferramenta OOPS! o próximo passo será corrigir as armadilhas apresentada na avaliação e reavaliar a ontologia de modo geral, principalmente os pontos que a ferramenta não está automatizada.

3.6 Considerações Finais

...

Tabela 3.2: Classes and Sub-classes of ontology prototype

Class	Sub-class	Sub-class
Documentation	Abstract	Keyword
		Limitations
		Conclusions
		Results
		Method
	Introduction	Goal
		Context
		Research Objective (GQM)
		Issue Issues
		Related Studies
	Related Work	Technology under Investigation
		Alternative Technology
		Relevance in Practice
		Observation
Planning	Template	
	References	
	Acknowledgment	
	Conclusion	
	Appendix	
	SPL Domain	
	Context	Local
	Experimental Unit	Selection of Participants
	Experimental Material	SPL artifacts
	Experimental Design	
	Goals	
	Hypotheses	
	Tasks	
	Parameters	
	Variables	
Execution	Procedures	
	Analysis of procedures	
	Preparation	
Analyze	Execution	Pilot
	Deviation from Planning	
	Type of Analysis	Type of Hypothesis Analysis
Discussion	Hypothesis Testing	
	Descriptive statistics	
	Qualitative analysis	
	Preparing the Dataset	
	Threats to Validity	Threat to Validity in SPL
	Evaluation of Results	
	Inferences	
Packaging		
Evaluation		

Tabela 3.3: Result Summary of the OOPS! evaluation!

Pitfall	Description	Critical Level
P08	Missing annotations in 119 cases	Minor
P10	Missing disjointness on ontology*	Important
P12	Equivalent properties not explicitly declared in 1 case	Important
P13	Inverse relationships not explicitly declared in 8 cases	Minor
P19	Defining multiple domains or ranges in properties in 6 cases	Critical
P41	No license declared on ontology	Important

Avaliação da SMartOntology: Um estudo Quantitativo

4.1 Considerações Iniciais

Este capítulo apresenta um estudo quantitativo realizado para avaliar a SMartOntology proposta no Capítulo 3, com base em um questionário contendo onze questões, oito com critérios de qualidade no estilo escala Likert /citeLikert, uma avaliação de frequência da escala Likert, teste estatístico e sua correlação entre os oito critérios.

Os oito critérios de qualidade foram identificados e definidos na Parte I da tese *Ontology Evaluation* de /citeDenny Vrandecic, são eles:

- **Precisão:** um critério que determina se os axiomas da ontologia estão em conformidade com o conhecimento das partes interessadas sobre o domínio. Uma maior precisão vem das definições e descrições corretas de classes, propriedades e indivíduos. A correção neste caso pode significar conformidade com ?padrões-ouro? definidos, sejam outras fontes de dados, conceituações ou mesmo realidade. /cite(Ceusters e Smith (2006)), introduz uma abordagem para usar a realidade como referência, ou seja, os termos da ontologia captura as partes pretendidas da realidade. Os axiomas devem restringir as possíveis interpretações de uma ontologia para que os modelos resultantes sejam compatíveis com as conceituações dos usuários;
- **Adaptabilidade:** mede até que ponto a ontologia antecipa seus usos. Uma ontologia deve oferecer a base conceitual para uma série de tarefas antecipadas

(idealmente, na Web, também deve oferecer a base para tarefas nunca antecipadas). Deve ser possível estender e especializar a ontologia monotonicamente, ou seja, sem a necessidade de remover axiomas (observe que em OWL, a monotonicidade semântica é dada pela monotonicidade sintática, ou seja, para retrair inferências, axiomas explícitos e implícitos precisam ser retraídos). Uma ontologia deve reagir de forma previsível e intuitiva a pequenas mudanças nos axiomas. Deve permitir metodologias para extensão, integração e adaptação, ou seja, incluir metadados necessários. Novas ferramentas e situações inesperadas devem poder usar a ontologia;

- **Clareza:** mede com que eficácia a ontologia comunica o significado pretendido dos termos definidos. As definições devem ser objetivas e independentes do contexto. Os nomes dos elementos devem ser compreensíveis e inequívocos. Uma ontologia deve usar definições em vez de descrições para classes. As entidades devem ser documentadas o suficiente e estar totalmente rotuladas em todos os idiomas necessários. Axiomas complexos devem ser documentados. As escolhas de representação não devem ser feitas para a conveniência da notação ou implementação, ou seja, o viés de codificação deve ser minimizado;
- **Completeness:** O critério Completeness mede se o domínio de interesse está coberto adequadamente. Todas as perguntas que a ontologia deve poder responder podem ser respondidas. Existem diferentes aspectos da completeness: completeness no que diz respeito ao idioma (tudo o que é declarado que poderia ser declarado usando o idioma fornecido?), Completeness no que diz respeito ao domínio (todos os indivíduos estão presentes, todos os conceitos relevantes são capturados?), Completeness com relação aos requisitos de aplicação (todos os dados necessários estão presentes?), etc. A abrangência também abrange a granularidade e a riqueza da ontologia;
- **Eficiência Computacional:** mede a capacidade das ferramentas usadas de trabalhar com a ontologia, em particular a velocidade que os raciocinadores precisam para executar as tarefas necessárias, seja resposta de consulta, classificação ou verificação de consistência. Alguns tipos de axiomas podem causar problemas para certos raciocinadores. O tamanho da ontologia também afeta a eficiência da ontologia;
- **Concisão:** determina se a ontologia inclui elementos irrelevantes em relação ao domínio a ser coberto (ou seja, uma ontologia sobre livros, incluindo axiomas sobre leões africanos) ou representações redundantes da semântica. Uma ontologia deve impor um compromisso ontológico mínimo, ou seja, especificar a teoria mais fraca possível. Somente termos essenciais devem ser definidos. As suposições subjacentes

da ontologia sobre o domínio mais amplo (especialmente sobre a realidade) devem ser tão fracas quanto possível, a fim de permitir a reutilização interna e a comunicação entre as partes interessadas que se comprometem com diferentes teorias;

- **Consistência:** mede que a ontologia não inclui ou permite contradições. Enquanto a precisão declara a conformidade da ontologia com uma fonte externa, a consistência indica que a própria ontologia pode ser interpretada. A consistência lógica é apenas uma parte dela, mas também as descrições formais e informais na ontologia devem ser consistentes, ou seja, a documentação e os comentários devem estar alinhados com os axiomas. Outros princípios de ordenação podem ser definidos com os quais a ontologia deve ser consistente, como as restrições OntoClean à taxonomia (Guarino e Welty, 2002); e
- **Capacidade Organizacional:** agrega vários critérios que decidem com que facilidade uma ontologia pode ser implantada dentro de uma organização. Ferramentas, bibliotecas, fontes de dados e outras ontologias usadas restringem a ontologia, e a ontologia deve atender a essas restrições. As ontologias são frequentemente especificadas usando uma metodologia de engenharia de ontologia ou usando conjuntos de dados específicos. Os metadados da ontologia podem descrever as metodologias, ferramentas e fontes de dados aplicadas e a organização. Esses metadados podem ser usados pela organização para decidir se uma ontologia deve ser aplicada ou não.

4.2 Escala Likert

A tese de /citeDenny Vrandecic não expõem uma forma clara para mensurar os oito critérios. Neste ponto foi definido que uma escala Likert poderia auxiliar. Segundo /citeLikert uma escala Likert para gerar bons resultados deve possuir três elementos fundamentais, (i) um negativo, (ii) um neutro e (iii) um positivo.

Os itens Likert são usados para medir as atitudes dos entrevistados em relação a uma pergunta ou afirmação específica. Essas escalas permitem determinar o nível de concordância ou discordância dos respondentes.

Dessa forma, a escala Likert pressupõe que a força e a intensidade da experiência são lineares. Portanto passa de uma concordância total a uma discordância total, assumindo que as atitudes podem ser medidas.

As respostas podem ser oferecidas em diferentes níveis de medição, permitindo escalas de 5, 7 e 9 elementos previamente configurados. Para analisar os dados, codificados esses itens da seguinte forma:

- 1: Discordo totalmente;
- 2: Discordo parcialmente;
- 3: Nem concordo nem discordo (neutro);
- 4: Concorde parcialmente; e
- 5: Concorde totalmente.

4.3 Definição do Estudo

O propósito principal deste estudo foi avaliar a SMartOntology proposta no Capítulo 3. Baseado no modelo Goal-Question-Metric (GQM) (Basili e Rombach, 1988), este estudo tem por objetivo:

Avaliar a SMartOntology proposta

Com o propósito de definir valores quantitativo

Referente os critérios de avaliação

Do ponto de vista de especialistas de ES, LPS e Ontologia

No contexto de pesquisadores das seguintes instituições: UEL, ICMC-USP, PUC-PR, UEM, UTFPR, USP, SIDIA, PUC-RS e Unipar.

4.4 Planejamento do Estudo

O planejamento do estudo é composto das seguintes etapas:

- **Projeto Piloto:** com a intenção de avaliar a instrumentação do estudo, um projeto piloto foi conduzido em Outubro de 2019, com um mestre e Doutor da Universidade Estadual de Maringá (UEM). Os dados obtidos foram descartados, mas, as considerações sobre erros e melhorias nos questionários foram consideradas.
- **Treinamento:** por causa do nível de conhecimento dos participantes não foi necessária a realização desta etapa.
- **Especialistas:** os especialistas convidados foram pesquisadores da área de ES, LPS e Ontologia. Dos participantes, X são pós-doutor (X%), X são doutores (X%), X são mestres (X%), X é mestrando (X%).

- **Instrumentação:** todos os especialistas receberam os seguintes documentos: i) um e-mail contendo um breve descritivo da avaliação, com os link para ferramentas e tecnologias necessárias; (ii) uma cópia do Instrumento de Avaliação Quantitativo caracterizado por um questionário - Google Forms; e (iii) uma cópia dos metadados da ontologia, os metadados são compostos por: um arquivo do tipo OWL do modelo da SMartOntology, um arquivo do tipo OWL da SMartOntology povoada com mais de 170 indivíduos, um arquivo do tipo planilha do Excel contendo os dados originais dos experimentos, uma cópia do artigo *An Ontology for Software Product Line Experiments* não publicado até o momento da escrita desta dissertação, uma cópia do capítulo 3 desta dissertação, uma cópia da tese de *Ontology Evaluation* de /cite-Denny Vrandecic, uma cópia do resumo da tese, uma cópia da Figura(XX)/diagrama de classe, uma cópia da Figura(XX)/modelo conceitual clusterizado, uma cópia da Figura(XX)/grafo inicial da ontologia, uma cópia da Figura(XX)/grafo gerado pela OWLweb.

A instrumentação fornecida aos especialistas foi suficiente para responder o questionário da avaliação.

4.5 Execução do Estudo

Esta seção apresenta as etapas seguidas durante a execução deste estudo.

Procedimentos de Participação: a participação de cada especialista no estudo ocorreu da seguinte forma:

1. o especialista recebe os documentos, via e-mail, que compõem o estudo (Seção 4.4);
2. o especialista faz um estudo prévio dos metadados, esclarece possíveis dúvidas;
3. o especialista faz a leitura e preenche o formulário de Avaliação Quantitativa - Google Forms, de acordo com as suas experiências;

Os especialistas tiveram um prazo de 40 dias para a finalização do questionário de avaliação. Todo o acompanhamento do estudo foi realizado online.

4.6 Análise e Discussão dos Resultados

Esta seção apresenta a análise e discussão dos resultados com relação às seguintes informações:

- perfil dos especialistas;
- apresentação da frequência da escala Likert;
- apresentação e análise do teste normalidade dos resultados;
- apresentação e análise do teste estatístico aplicado aos resultados;
- apresentação e análise da correlação entre os critérios de qualidade; e
- pontos de melhorias apontados pelos especialistas.

4.6.1 Perfil dos Especialistas

4.6.2 Frequência Likert

4.6.3 Teste de Normalidade

4.6.4 Teste Estatístico

4.6.5 Correlação dos Critérios

4.6.6 Melhorias

4.7 Ameaças à Validade

4.7.1 Ameaças à Validade Interna

- **Diferenças entre os especialistas:** devido do tamanho da amostra, as variações entre as habilidades dos especialistas foram poucas, assim, os especialistas não foram divididos em grupos. Dessa forma, as tarefas designadas foram idênticas;
- **Acurácia das respostas dos participantes:** uma vez que as informações referentes da SMartOntology são apresentadas juntamente com seus metadados e considerando que os participantes são especialistas em ES, LPS e Ontologia, considera-se que as respostas fornecidas são válidas;
- **Efeito de Fadiga:** os arquivos da SMartOntology (modelagem e povoados com indivíduos) são de difícil leitura pelos seres humanos, logo se faz necessária a utilização de ferramentas para exibição da SMartOntology, bem como os textos auxiliares, para uma maior compreensão dos critérios de avaliação, e das imagens

anexadas ao questionário. Os efeitos de fadiga são uma das principais ameaças, por isso, para tentar minimizar esse problema, a documentação foi enviada aos especialistas por e-mail com um prazo de 40 dias para enviarem as respostas, assim, eles poderiam responder conforme a sua disponibilidade;

- **Influência de outros especialistas na avaliação:** a influência entre os especialistas foi mitigada, pois eles não sabiam quem foram convidados para tal estudo; e
- **Análise dos dados:** a análise dos dados dos especialistas foi realizada somente pelo autor desta dissertação, sem a realização de uma revisão. Dessa forma, tal análise deve ser revisada em trabalhos futuros.

4.7.2 Ameaças à Validade Externa

A obtenção de especialistas qualificados nas área de ES, LPS e Ontologia foi uma das dificuldades encontradas neste estudo e muitos especialistas convidados não puderam participar do estudo por conta dos seus compromissos. Apesar de poucos especialistas participarem do estudo, a qualidade do perfil deles é a variável mais importante nessa avaliação.

4.7.3 Ameaças à Validade de Constructo

A instrumentação foi avaliada e adequada, conforme projeto piloto realizado. Quanto ao nível de conhecimento dos especialistas em ES, LPS e Ontologia demonstraram satisfatórios para avaliarem as diretrizes propostas. Os especialistas que não tinha familiaridade com Ontologia, pode encontrar no material de apoio um respaldo para execução das tarefas.

4.7.4 Ameaças à Validade de Conclusão

A principal ameaça para este estudo está relacionada ao número de especialistas (??). Este número é pequeno, mas o conhecimento dos especialistas que colaboraram com este estudo foi significativo. Além disso, pela característica própria do estudo por prezar a qualidade dos especialistas, muitas vezes o número baixo de especialistas pode ser considerado. Assim, não é possível generalizar as conclusões.

4.8 Empacotamento e Compartilhamento

Os documentos disponibilizados aos participantes e resultados deste estudo qualitativo estão disponíveis em: <https://doi.org/...>, visando disseminação dos dados.

4.9 Propostas de Melhoria para SMartOntology

4.10 Considerações Finais

Este capítulo apresentou a avaliação quantitativa sobre oito critérios de qualidade das diretrizes aplicados a SMartOntology com ?? especialistas em ES, LPS e Ontologia. Os resultados obtidos permitiram identificar a qualidade da SMartOntology. Assim, os resultados dos dados analisados forneceu evidências que a SMartOntology possui um grau de qualidade relevante. Permitindo assim a evolução de novos estudos sobre a SMartOntology, expandindo-a e melhorando ainda mais seus pontos de baixa qualidade.

Foram sugeridas melhorias à SMartOntology, em que devem ser aplicadas em trabalhos futuros. Por fim, compreende-se que novos estudos devem ser realizados com outros especialistas para expandir o corpo de conhecimento.

Um Sistema de Recomendação para Experimentos em LPS (Prova de Conceito) - RecSysExper-SPL

5.1 Considerações Iniciais

Este tópico apresenta uma prova de conceito por meio de um sistema de recomendação aplicando inferência sobre a SMartOntology.

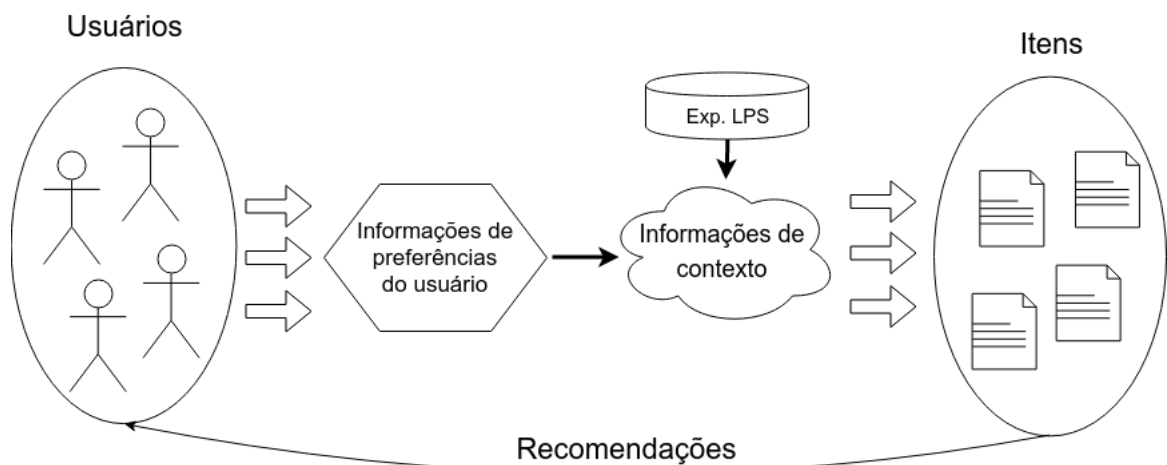


Figura 5.1: Modelagem geral da RSSE proposta

A Figura - 5.1 apresenta o conceito geral da metodologia deste projeto. Iniciando pela entrada dos usuários, depois extraímos as preferências dele, e em seguida será feita a

extração de informações de contexto utilizando a base de meta dados em LPS, para então fazer inferência nos itens (que são experimentos em LPS), por meio dessa inferência será obtido as recomendações.

Inicialmente, a entrada de dados dos usuários está sendo definido da seguinte forma:

- Informações de LPS:

Dominio;

Sub-dominio;

Tipos de artefatos e;

Feature module.

- Experimentos

filtrar por qualidade.

5.2 Sistema de Recomendação

Os sistemas de recomendação são aplicativos de software que visam dar suporte para usuário na tomada de decisões ao interagir com grandes espaços de informação. Estes softwares recomendam itens de interesse para os usuários com base em preferências que tenham sido expressas explicitamente ou implicitamente (Ricci et al., 2011). Segundo Mahmood e Ricci (2009) os sistemas de recomendação são técnicas ou ferramentas de software, que podem reduzir a sobrecarga de informações para os usuários, sugerindo itens, conteúdos ou serviços, entre outros.

Os sistemas de recomendação surgiram nos trabalhos extensivos das ciências cognitivas, teoria de aproximação, recuperação da informação e teoria de previsões e também possuem influências das ciências de administração e marketing (Allen e Fildes, 2001; Murthi e Sarkar, 2003). O primeiro sistema de recomendação proposto foi o *Tapestry*, nesse sistema criou-se um modelo mais usados em sistemas de recomendação, onde a recomendação de conteúdo é auxiliada pela colaboração de um grupo de pessoas, batizada como "filtragem colaborativa". Neste trabalho, iniciou o desafio de casar corretamente os que dados recomendados com os usuários que o recebem, analisando o real relacionamento de interesse (Kwong et al., 1992; Resnick e Varian, 1997).

Existem cinco abordagens mais usadas em sistemas de recomendação, três tradicionais: Filtragem Colaborativa (*Collaborative Filtering*), Filtragem Baseada em Conteúdo (*Content-based Filtering*) e Recomendação Baseada no Conhecimento (*Knowledge-Based*

Recommendation), e duas modernas: Sistemas de Recomendação Híbridos (*Hybrid Recommender Systems*) e Sistemas de Recomendação usando Informações de Contexto (*Context-aware Recommender Systems*).

Dado que este trabalho trata o sistema de recomendação como uma prova de conceito para realização de inferência sobre a SMartOntology, estará descrito a seguir somente o modelo implementado no projeto de sistema de recomendação.

Collaborative Filtering

A Filtragem Colaborativa baseia-se na idéia de "boca-a-boca" em que a informação passada de pessoa a pessoa desempenha um papel importante ao tomar uma decisão. Abstraindo, as pessoas são substituídas pelos chamados vizinhos mais próximos (NN) que são usuários com um padrão de preferência ou comportamento semelhante ao usuário atual. (Robillard et al., 2010). Filtragem Colaborativa depende de dois tipos diferentes de dados: (1) um conjunto de usuários e (2) um conjunto de itens. A relação entre usuários e itens é expressada principalmente em termos de *ratings* fornecidos pelos usuários e explorados em futuras sessões de recomendação para prever a classificação de um usuário (Robillard et al., 2010).

Em Engenharia de Software (*Recommendation System in Software Engineering - RSSEs*), sistemas de recomendação desempenham importantes funções a fim de ajudar a equipe de software a lidar com sobrecarga de informações, filtrando e fornecendo informações úteis. São ferramentas de software introduzidas especificamente para ajudar equipes de desenvolvimento de software e partes interessadas a lidar com a busca de informações em um determinado contexto em ES (Robillard et al., 2010).

Robillard et al. (2010) comenta que, em um ambiente de desenvolvimento de software aplicando ES existe um *landscape* de informações sobre o projeto em desenvolvimento, e este espaço de informações pode ser categorizados por:

- Código fonte do projeto;
- História do projeto;
- Arquivos de comunicação;
- Dependências de API em outras fontes;
- Ambiente de desenvolvimento;
- Logs de interação entre os usuários;

- Logs de execução e;
- A web.

Um RSSE pode trazer simultaneamente dois aspectos distintos: (i) novidade e surpresa, porque as RSSEs ajudam a descobrir novas informações e (ii) traz familiaridade e reforço, pois as RSSEs suportam a confirmação do conhecimento existente. Finalmente, referenciar uma tarefa e um contexto específicos, distingue RSSEs de ferramentas de pesquisa genéricas, por exemplo, uma ferramentas de RSSR para ajudar os desenvolvedores a encontrar exemplos de código fonte (Robillard et al., 2010).

RSSE compreende três componentes principais, (i) um mecanismo para coletar dados, (ii) um mecanismo de recomendação para analisar dados e gerar recomendações e (iii) uma interface de usuário para fornecer recomendações (Rahman et al., 2014).

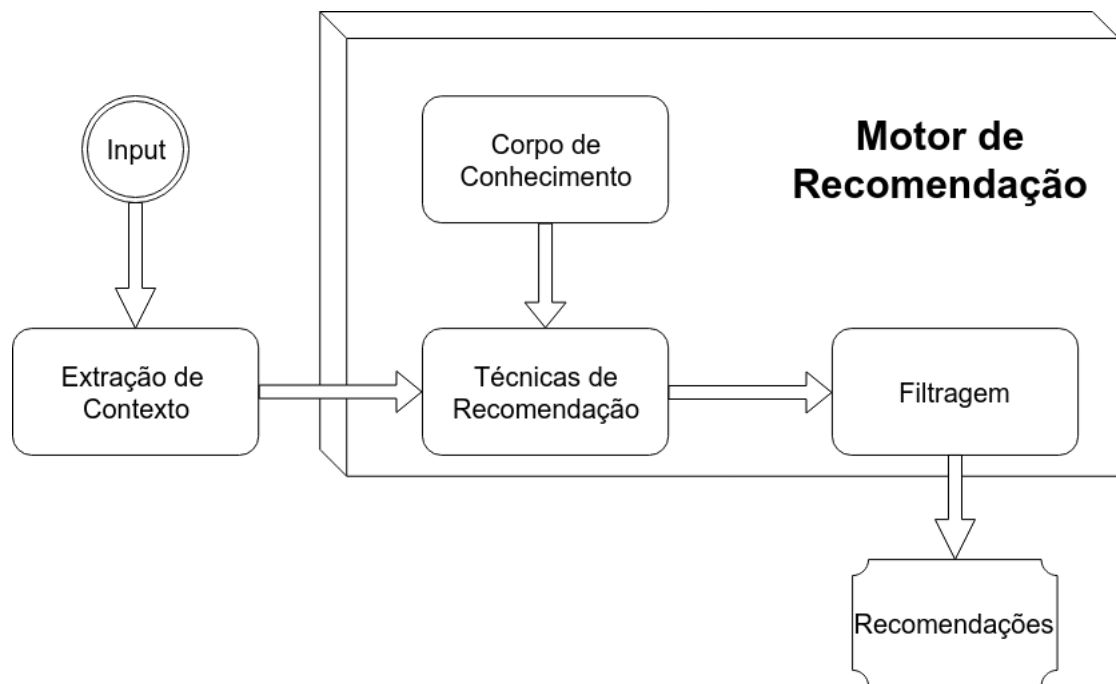


Figura 5.2: Passos de Construção para um RSSE. Traduzido e Adaptado de Maki (2016)

A Figura - 5.2 apresenta de forma geral como é construído um RSSE, partindo da entradas dos dados pelo *input*, passando pela extração de contexto, seguindo para aplicação de alguma técnica de recomendação, na qual sofre um inferência do corpo de conhecimento (normalmente específico para cada área de ES), depois segue para um processo de filtragem dos resultados, e como saída a recomendação em si.

Foi encontrado uma revisão sistemáticas (trabalho da Maki, 2016), que aborda métodos e modelos de implementação de um RSSE apresentando vários aspectos de SR em ES,

principalmente no tipo de corpo de conhecimento aplicado a RSSE. Nessa revisão foi possível identificar algumas áreas da ES que utiliza SR, apresentadas a seguir.

- SR para exploração código fonte;
- SR para reuso de software;
- SR para refatoração de código fonte (por exemplo, *class* em POO);
- SR para reuso de componentes de software;
- SR na exploração de APIs;
- SR na depuração de código (*debugging*)
- SR na recomendação de agentes *Agile*
- SR na descoberta de requisitos;
- SR na mudança do ciclo de vida;
- SR na evolução do ciclo de vida e;
- SR na busca de *bugs*.

5.3 Concepção

5.4 Projeto

5.5 Ambiente de Desenvolvimento

5.6 Interface com Usuário

5.7 Considerações Finais

Inicialmente foi apresentados os modelos básico de sistemas de recomendação tradicionais, como *collaborative filtering* e *content-based filtering*. Na sequência foi apresentado como os sistemas de recomendação são tratados na ES e alguns exemplos de aplicação.

Conclusão

6.1 Considerações Iniciais

...

A realização de experimentos de SE no SPL traz informações mais relevantes para fornecer evidências de uma teoria para o mundo real. A capacidade de entender, estudar e replicar experimentos torna esse método ainda mais rico. Ao organizar os dados e informações sobre os experimentos de SE no SPL, acreditamos que essa tarefa é mais fácil. Usando as tecnologias adjacentes, tais como formalizar o conhecimento através de ontologias e inferências para gerar dados de informação personalizada através de um sistema de recomendação.

Neste trabalho, foi desenvolvida uma proposta para um modelo de ontologia (SMartyOntology) com o objetivo de organizar e estruturar o conhecimento adquirido sobre experimentos de SE em SPL. Este conhecimento foi previamente levantado em um trabalho de dissertação do nosso grupo de pesquisa, onde foram relatados mais de 200 artigos que relatam experimentos em SPL. Com este levantamento foi possível criar um modelo de ontologia para representar o conhecimento sobre experimentos em NPS, inserindo os dados dos artigos neste modelo, denominamos esta fase de inserção dos indivíduos na ontologia. Desta forma, estruturamos o TBox e o Abox para a SMartyOntology. Com a composição da ontologia podemos fazer uma breve avaliação sobre algumas armadilhas que o modelo de ontologia poderia ter, utilizamos a ferramenta OOPS! para esta avaliação. Também foi possível criar um exemplo simples de recomendação, fazendo a inferência das informações para a SMartyOntology.

A SMartyOntology se destaca por levar em conta um domínio específico de experimentos de SE. SPL são construídos através de um domínio de aplicação, semelhanças, avaliações do núcleo e variabilidades, que distingue um produto do outro dentro da família de produtos, todas essas características estão presentes na ontologia. No entanto, a SMartyOntology pode ser facilmente estendida a outros domínios de experimentos de SE, uma vez que todas as classes dentro da ontologia que lidam com SPL são subclasses de representações de experimentos em geral.

Os principais objetivos deste trabalho, estão relacionados a propor um sistema de recomendação que, possa gerar processos e diretrizes para realização de experimentos em LPS. Para foi realizado um estudo aprofundado nos conceitos de Sistemas de Recomendação em ES e modelos de Ontologias para representação dos dados levantados sobre a qualidade dos experimentos em LPS, encontrados no trabalho do grupo.

Portanto, acredita-se que após a realização do projeto de software, será possível desenvolver o sistema de recomendação para que os usuários que interajam com este sistema, e possa receber recomendações de processos e diretrizes para suas pesquisas experimentais em LPS.

Como trabalho futuro, identificamos na avaliação vários pontos de melhoria na modelagem da ontologia, esses ajustes devem ser feitos para padronizar o modelo com o objetivo de possibilitar a extensão e divulgação do mesmo.

6.2 Considerações Finais

...

REFERÊNCIAS

- ALLEN, P. G.; FILDES, R. Econometric forecasting. In: *Principles of forecasting*, Springer, p. 303–362, 2001.
- ALMEIDA, M. B.; BAX, M. P. Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. *Ciência da informação*, v. 32, n. 3, p. 7–20, 2003.
- APEL, K.-O. *Analytic philosophy of language and the geisteswissenschaften*. Springer, 2013.
- BLONDET, G.; LE DUIGOU, J.; BOUDAUD, N. Ode: an ontology for numerical design of experiments. *Procedia CIRP*, v. 50, p. 496–501, 2016.
- CALVANESE, D.; DE GIACOMO, G.; LEMBO, D.; LENZERINI, M.; ROSATI, R. Dl-lite: Tractable description logics for ontologies. In: *AAAI*, 2005, p. 602–607.
- DA CRUZ, S. M. S.; CAMPOS, M. L. M.; MATTOSO, M. A foundational ontology to support scientific experiments. In: *ONTOBRAS-MOST*, ACM: USA, 2012, p. 144–155.
- CRUZES, D.; MENDONCA, M.; BASILI, V.; SHULL, F.; JINO, M. Extracting information from experimental software engineering papers. In: *XXVI International Conference of the Chilean Society of Computer Science (SCCC'07)*, USA: IEEE, 2007, p. 105–114.
- DIESTE, O.; JURISTO, N. Systematic review and aggregation of empirical studies on elicitation techniques. *IEEE Transactions on Software Engineering*, v. 37, n. 2, p. 283–304, 2011.
- DIESTE, O.; JURISTO, N. Challenges of evaluating the quality of software engineering experiments. In: *Perspectives on the Future of Software Engineering*, Springer, p. 159–177, 2013.

- FERNANDA, M. Modelagem conceitual: a construção de uma ontologia sobre avaliação do ciclo de vida (acv) para fomentar a disseminação de seus conceitos. 2007.
- FURTADO, V. R. *Guidelines for software product line experiment evaluation*. Dissertação de Mestrado, State University of Maringá, Maringá-PR. Brazil, in Portuguese, 2018.
- GARCIA, R. E.; HÖHN, E. N.; BARBOSA, E. F.; MALDONADO, J. C. An ontology for controlled experiments on software engineering. In: *SEKE, USA: ACM*, 2008, p. 685–690.
- GELERNTER, J.; JHA, J. Challenges in ontology evaluation. *Journal of Data and Information Quality (JDIQ)*, v. 7, n. 3, p. 11, 2016.
- GRUBER, T. What is an ontology. *WWW Site <http://www-ksl.stanford.edu/kst/whatis-an-ontology.html>* (accessed on 07-09-2004), 1993.
- HALMANS, G.; POHL, K. Communicating the variability of a software-product family to customers. *Software and Systems Modeling*, v. 2, n. 1, p. 15–36, 2003.
- KAMPENES, V. Quality of design, analysis and reporting of software engineering experiments: A systematic review. 2007.
- KITCHENHAM, B.; BUDGEN, D.; BRERETON, P.; TURNER, M.; CHARTERS, S.; LINKMAN, S. Large-scale software engineering questions—expert opinion or empirical evidence? *IET software*, v. 1, n. 5, p. 161–171, 2007.
- KITCHENHAM, B. A.; BUDGEN, D.; BRERETON, P. *Evidence-based software engineering and systematic reviews*, v. 4. CRC Press, 2015.
- KWONG, K. K.; BELLIVEAU, J. W.; CHESLER, D. A.; GOLDBERG, I. E.; WEISSKOFF, R. M.; PONCELET, B. P.; KENNEDY, D. N.; HOPPEL, B. E.; COHEN, M. S.; TURNER, R. Dynamic magnetic resonance imaging of human brain activity during primary sensory stimulation. *Proceedings of the National Academy of Sciences*, v. 89, n. 12, p. 5675–5679, 1992.
- VAN DER LINDEN, F.; SCHMID, K.; ROMMES, E. The product line engineering approach. In: *Software Product Lines in Action*, Springer, p. 3–20, 2007.
- LOHMANN, S.; NEGRU, S.; HAAG, F.; ERTL, T. Visualizing ontologies with owl. *Semantic Web*, v. 7, n. 4, p. 399–419, 2016.

- MAHMOOD, T.; RICCI, F. Improving recommender systems with adaptive conversational strategies. In: *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, ACM, 2009, p. 73–82.
- MAKI, S. *Systematic review of recommendation systems in software engineering*. Tese de Doutorado, École de technologie supérieure, 2016.
- MCKINNEY, W. Data structures for statistical computing in python. In: *Proceedings of the 9th Python in Science Conference*, USA, 2010, p. 51 – 56.
- MURTHI, B.; SARKAR, S. The role of the management sciences in research on personalization. *Management Science*, v. 49, n. 10, p. 1344–1362, 2003.
- MUSEN, M. A. The protégé project: a look back and a look forward. *AI Matters*, v. 1, n. 4, p. 4–12, 2015.
Disponível em <https://doi.org/10.1145/2757001.2757003>
- NORTHROP, L.; CLEMENTS, P.; BACHMANN, F.; BERGEY, J.; CHASTEK, G.; COHEN, S.; DONOHOE, P.; JONES, L.; KRUT, R.; LITTLE, R.; ET AL. A framework for software product line practice, version 5.0. *SEI-2007-http://www.sei.cmu.edu/productlines/index.html*, 2007.
- PÉREZ, F.; GRANGER, B. E. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, v. 9, n. 3, p. 21–29, 2007.
Disponível em <https://ipython.org>
- POHL, K.; BÖCKLE, G.; VAN DER LINDEN, F. J. *Software product line engineering: foundations, principles and techniques*. Springer Science & Business Media, 2005.
- RAHMAN, M. M.; YEASMIN, S.; ROY, C. K. Towards a context-aware ide-based meta search engine for recommendation about programming errors and exceptions. In: *Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE), 2014 Software Evolution Week-IEEE Conference on*, IEEE, 2014, p. 194–203.
- RESNICK, P.; VARIAN, H. R. Recommender systems. *Communications of the ACM*, v. 40, n. 3, p. 56–58, 1997.
- RICCI, F.; ROKACH, L.; SHAPIRA, B. Introduction to recommender systems handbook. In: *Recommender systems handbook*, Springer, p. 1–35, 2011.

- ROBILLARD, M.; WALKER, R.; ZIMMERMANN, T. Recommendation systems for software engineering. *IEEE software*, v. 27, n. 4, p. 80–86, 2010.
- SCATALON, L. P.; GARCIA, R. E.; CORREIA, R. C. M. Packaging controlled experiments using an evolutionary approach based on ontology (s). In: *SEKE*, 2011, p. 408–413.
- SOLDATOVA, L. N.; KING, R. D. An ontology of scientific experiments. *Journal of the Royal Society Interface*, v. 3, n. 11, p. 795–803, 2006.
- SOMMERVILLE, I. Software engineering, boston, massachusetts: Pearson education. 2011.
- TEIXEIRA, E. O. Análise da qualidade de experimentos controlados no contexto da engenharia de software empírica. 2014.
- WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. *Experimentation in software engineering*. Springer Science & Business Media, 2012a.
- WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. *Experimentation in Software Engineering*. Springer Science & Business Media, 2012b.
- WOLFF, C.; ÉCOLE, J. *Philosophia prima sive ontologia*. 1962.

Apêndice A: Código Fonte do Modelo da SMartOntology

A.1 Saída Da Opção time-flags com os Breakpoints .dijkstra e .main

```

-----
- TOOLCHAIN UNB-RISC16 -
- Energy Profile Version 1.00 -
- -
- Trip Time Flags -
- Options: time-flags & BreakPoints: .dijkstra and .main
- - Sumary
- Program dijkstra.o
- -
- - 1-Times
- - 1.1 Real time
- - 1.2 Processor Time
- - 2-Flags Reached
-----
- Section 1 [Times]
-
- 1.1 Time Consumption - Real Time (in seconds): 0
- 1.2 Time Consumption - Processor UNB-RISC16 (in seconds): 0,000000688
-----

- Function .d Reached After 0 Cycles From the Last Begin
- Function .main Reached After 8 Cycles From the Last Point
-----

```

Apêndice B: Código Fonte do Povoamento de Dados no Modelo SMartOntology

Apêndice C: Questionário de Avaliação e Respostas Aplicado à SMartOntology

Apêndice D: Código Fonte dos Exemplos de Aplicação de Recomendação Usando SMartOntology
