

JOSELAINE VALASKI

DERIVAÇÃO DE REQUISITOS FUNCIONAIS A
PARTIR DE DESCRIÇÕES DE DOMÍNIO

Proposta de tese apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná para obtenção do título de Doutor em Informática.

Curitiba
2017

JOSELAINÉ VALASKI

DERIVAÇÃO DE REQUISITOS FUNCIONAIS A PARTIR DE DESCRIÇÕES DE DOMÍNIO

Proposta de tese apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná para obtenção do título de Doutor em Informática.

Área de concentração: Ciência da Computação

Orientadora: Profa. Andreia Malucelli
Coorientadora: Profa. Sheila Reinehr

Curitiba
2017

FICHA CATALOGRÁFICA

V137d Valaski, Joselaine
2017 Derivação de requisitos funcionais a partir de descrições de domínio /
Joselaine Valaski ; orientadora, Andreia Malucelli ; coorientadora, Sheila
Reinehr. -- 2017
186 f. : il. ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Paraná, Curitiba,
2017.
Bibliografia: f. 132-144

1. Informática. 2. Engenharia de software). 3. Software – Desenvolvimento.
4. Computação (Semântica). 5. UML (Computação). I. Malucelli, Andreia. II.
Reinehr, Sheila dos Santos. III. Pontifícia Universidade Católica do Paraná.
Programa de Pós-Graduação em Informática. IV. Título.

CDD 20. ed. – 004



Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Informática

ATA DE SESSÃO PÚBLICA

DEFESA DE TESE DE DOUTORADO Nº 47/2017

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA – PPGIa PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ - PUCPR

Em sessão pública realizada às 16h00 de 23 de Agosto de 2017, na Sala 208 – Escola de Negócios, ocorreu a defesa da tese de doutorado intitulada “Derivação de Requisitos Funcionais a partir de Descrições de Domínio” elaborada pela aluna **Joselaine Valaski**, como requisito parcial para a obtenção do título de **Doutor em Informática**, na área de concentração **Ciência da Computação**, perante a banca examinadora composta pelos seguintes membros:

Prof.ª Dr.ª Andreia Malucelli (orientadora) - PPGIa/PUCPR

Prof.ª Dr.ª Sheila dos Santos Reinehr (co-orientadora) – PUCPR

Prof. Dr. Emerson Cabrera Paraiso - PUCPR

Prof. Dr. Cesar Tacla – UTFPR

Prof. Dr. Paulo Cesar Stadzisz – UTFPR

Prof.ª Dr.ª Renata Guizzardi – UFES



Após a apresentação da tese pelo aluno e correspondente arguição, a banca examinadora emitiu o seguinte parecer sobre a tese:

Membro	Parecer
Prof.ª Dr.ª Andreia Malucelli	(X) Aprovada () Reprovada
Prof.ª Dr.ª Sheila dos Santos Reinehr	(X) Aprovada () Reprovada
Prof. Dr. Emerson Cabrera Paraiso	(X) Aprovada () Reprovada
Prof. Dr. Cesar Tacla	(X) Aprovada () Reprovada
Prof. Dr. Paulo Cesar Stadzisz	(X) Aprovada () Reprovada
Prof.ª Dr.ª Renata Guizzardi	(X) Aprovada () Reprovada

Portanto, conforme as normas regimentais do PPGIa e da PUCPR, a tese foi considerada:


(X) APROVADO


(aprovação condicionada ao atendimento integral das correções e melhorias recomendadas pela banca examinadora, conforme anexo, dentro do prazo regimental)

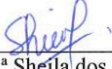
() REPROVADO


E, para constar, lavrou-se a presente ata que vai assinada por todos os membros da banca examinadora. Curitiba, 23 de Agosto de 2017.



Prof.ª Dr.ª Andreia Malucelli


Prof. Dr. Cesar Tacla


Prof.ª Dr.ª Renata Guizzardi


Prof.ª Dr.ª Sheila dos Santos Reinehr


Prof. Dr. Paulo Cesar Stadzisz


Prof. Dr. Emerson Cabrera Paraiso

DEDICATÓRIAS

Aos meus pais Floriano e Emília.

Ao Sergio Nunes.

À Julia, meu maior motivo de viver.

AGRADECIMENTOS

Aos meus pais Floriano e Emília, meus queridos, pelo amor, companheirismo e por sempre estar ao meu lado mesmo em situações inesperadas.

Ao Sergio Nunes, meu maior incentivador, sempre me encheu de palavras de otimismo e motivação. Com certeza, você é a pessoa que mais acredita no meu potencial e que mais me leva para frente. Sergio, você é o melhor companheiro que eu poderia ter.

Aos amigos do grupo de pesquisa, por compartilhar momentos bons, mas também momentos difíceis e desta maneira tornar a caminhada até aqui mais leve.

Aos professores de todos os níveis de ensino que transmitiram seu saber, sua experiência, sua bondade e me motivam a ser um deles.

À Fernanda Baião e sua equipe sempre disponíveis e com contribuições valiosas.

Ao casal Guizzardi que me atenderam em diversos momentos e me ajudaram em diversos pontos desta caminhada.

À Sheila Reinehr por me aceitar no seu grupo, por compartilhar tanta experiência e conhecimento e pelas orientações e indicações sempre muito relevantes.

À Andreia, por aceitar novamente o desafio de ser minha orientadora, apesar de já conhecer meus defeitos no mestrado. Andreia, você já transcendeu o papel de orientadora, você é minha amiga, minha inspiração e fonte de muita diversão, aconteça o que acontecer.

*"A boa educação é moeda de ouro, em toda parte tem valor."
(Padre Antônio Vieira)*

RESUMO

No processo de desenvolvimento de software, a área de Engenharia de Requisitos fornece os mecanismos para analisar as reais necessidades do cliente. A primeira atividade prevista neste processo é a elicitación de requisitos. A elicitación de requisitos é uma atividade orientada para o entendimento do domínio do software a ser desenvolvido e, portanto, demanda qualidade nas comunicações entre os envolvidos. Requisitos de software mal interpretados podem ter consequências negativas para todas as demais etapas do processo de desenvolvimento de software. Embora existam diversas técnicas para a elicitación de requisitos, elas ainda são bastante informais e o sucesso delas depende da experiência dos recursos humanos envolvidos. Alguns problemas comuns nesta atividade são: comunicação deficiente, falta de consenso sobre os termos utilizados e falta de conhecimento do domínio a ser elicitado. Neste sentido, um modelo conceitual pode ser um instrumento importante para facilitar a comunicação entre os envolvidos e explicitar conhecimentos de um domínio. No entanto, o modelo conceitual deve ser representado em uma linguagem expressiva que potencialize os benefícios de um modelo conceitual. A OntoUML é uma linguagem ontologicamente fundamentada que possibilita maior expressividade do que outras linguagens não fundamentadas ontologicamente. No entanto, a construção de um modelo conceitual ontologicamente fundamentado, não é uma tarefa trivial. Esta dificuldade ocorre pela complexidade da atividade de construção de um modelo conceitual e também pelo pouco conhecimento da OntoUML pelos profissionais. Neste contexto, esta pesquisa propõe uma abordagem centrada em modelos conceituais OntoUML para apoiar a derivação de requisitos funcionais a partir de descrições de domínio. A abordagem é apoiada por métodos e ferramentas. Esta pesquisa foi realizada de acordo com cinco etapas principais: exploração do objeto, avaliação da expressividade da OntoUML, proposta de método para a derivação dos requisitos funcionais, desenvolvimento do ambiente computacional e avaliação da abordagem proposta. Os resultados apontaram que a abordagem apoia significativamente a identificação semiautomática dos construtos OntoUML, bem como a derivação de requisitos funcionais de domínio, a partir de um modelo conceitual em OntoUML.

Palavras-chaves: Engenharia de Requisitos, OntoUML, Modelo Conceitual, Semântica.

ABSTRACT

In the software development process, the requirement engineering area provides the mechanisms to analyze the client's real needs. The first activity foreseen in this process is requirement elicitation. Requirement elicitation is an activity aimed at understanding the domain of the software to be developed and therefore, it requires quality in the communication among the stakeholders. Misinterpreted software requirements may have negative consequences to all of the remaining stages of software development. Although there are several requirements elicitation techniques, they are still quite informal and their success depend on the human resources involved in the process. Some common issues found in this activity are as follows: poor communication, lack of consensus regarding the used terms and lack of knowledge about the elicited domain. In this sense, a conceptual model may be an important instrument in order to facilitate the communication among the stakeholders and to make explicit the domain knowledge. However, the conceptual model must be represented in an expressive language, which can enable one to profit from the benefits of a conceptual model. OntoUML is an ontologically well-founded language which is more expressive than others, not ontologically well-founded languages. However, the construction of an ontologically well-founded conceptual model is not a trivial task. This difficulty occurs due to the complexity of the conceptual model development activity and due to the lack of knowledge of OntoUML practitioners. In this context, this research proposes an approach centered on OntoUML conceptual models in order to support the derivation of functional requirements from domain descriptions. The approach is supported by methods and tools. This research has been performed according to five main stages: object exploration, OntoUML expressivity evaluation, method proposal for functional requirement derivation, computational environment development and evaluation of proposed approach. The results demonstrate that the approach significantly supports the semi-automated identification of OntoUML constructs, as well as the derivation of functional requirements of a domain from a OntoUML conceptual model.

Keywords: Requirements Engineering, OntoUML, Conceptual Model, Semantic.

SUMÁRIO

RESUMO	VIII
ABSTRACT	IX
LISTA DE FIGURAS	XIV
LISTA DE TABELAS	XVI
LISTA DE QUADROS.....	XVII
LISTA DE ABREVIATURAS E SIGLAS.....	XVIII
CAPÍTULO 1 - INTRODUÇÃO.....	1
1.1 MOTIVAÇÃO.....	4
1.2 OBJETIVOS	7
1.3 DELIMITAÇÃO DE ESCOPO.....	7
1.4 ESTRUTURA DO DOCUMENTO DA TESE	8
1.5 CONSIDERAÇÕES SOBRE O CAPÍTULO.....	10
CAPÍTULO 2 - ENGENHARIA DE REQUISITOS E ONTOLOGIAS	11
2.1 ENGENHARIA DE REQUISITOS DE SOFTWARE	11
2.1.1 Elicitação de requisitos	12
2.1.2 Técnicas de elicitação de requisitos	13
2.2 ONTOLOGIAS.....	16
2.2.1 Definição de ontologia	17
2.2.2 Tipos de ontologias.....	18
2.2.3 Construção de ontologias	20
2.3 CONSIDERAÇÕES SOBRE O CAPÍTULO.....	21
CAPÍTULO 3 - USO DAS ONTOLOGIAS NA ENGENHARIA DE REQUISITOS: REVISÃO SISTEMÁTICA	22
3.1 MOTIVAÇÃO.....	22
3.2 MÉTODO DE PESQUISA.....	23
3.2.1 Planejamento da revisão	23
3.2.2 Identificação da pesquisa	23
3.2.3 Seleção de estudos primários	24
3.2.4 Classificação.....	25
3.3 RESULTADOS	27

3.3.1	QP1: Quais são as atividades da ERS nas quais as ontologias têm sido aplicadas?.....	27
3.3.2	QP2: Quais são as funções das ontologias na ERS?	28
3.3.3	QP3: Quais são as linguagens usadas pelas ontologias?.....	29
3.3.4	QP4: Quais são os domínios do conhecimento representados?.....	31
3.3.5	QP5: Quais são as contribuições das ontologias para a ERS?.....	33
3.4	CONSIDERAÇÕES SOBRE A REVISÃO SISTEMÁTICA	35
3.5	CONSIDERAÇÕES SOBRE O CAPÍTULO.....	37
CAPÍTULO 4 - ONTOUML		38
4.1	OBJETIVO	38
4.2	CONSTRUTOS: SUBSTANTIAL UNIVERSAL.....	39
4.2.1	Sortal Universal	40
4.2.2	Mixin Universal	43
4.3	CONSTRUTOS: MOMENT UNIVERSAL E RELATION	46
4.4	CONSIDERAÇÕES SOBRE O CAPÍTULO.....	51
CAPÍTULO 5 - ESTRUTURAÇÃO DA PESQUISA		52
5.1	CARACTERIZAÇÃO DA PESQUISA	52
5.2	ESTRATÉGIA DA PESQUISA.....	52
5.2.1	Exploração do objeto	54
5.2.2	Avaliação da expressividade da OntoUML.....	54
5.2.3	Proposta de método para a derivação de requisitos funcionais a partir de modelos conceituais em OntoUML	55
5.2.4	Desenvolvimento de ambiente para a construção de modelo conceitual em OntoUML.....	55
5.2.5	Avaliação da abordagem proposta	57
5.3	CONSIDERAÇÕES SOBRE O CAPÍTULO.....	59
CAPÍTULO 6 - AVALIAÇÃO DA EXPRESSIVIDADE DA ONTOUML		60
6.1	FASES DA AVALIAÇÃO	60
6.1.1	Seleção do domínio	61
6.1.2	Construção do modelo conceitual em OntoUML	61
6.1.3	Construção do modelo conceitual em UML.....	62
6.1.4	Avaliação da expressividade dos modelos.....	62
6.2	RESULTADOS E DISCUSSÃO	63
6.2.1	Modelo conceitual em OntoUML	63
6.2.2	Modelo conceitual em UML	64

6.2.3	Avaliação da expressividade dos modelos conceituais construídos	65
6.3	CONSIDERAÇÕES SOBRE A AVALIAÇÃO	69
6.4	CONSIDERAÇÕES SOBRE O CAPÍTULO	70
CAPÍTULO 7 - MÉTODO PARA A DERIVAÇÃO DE REQUISITOS FUNCIONAIS DE DOMÍNIO 71		
7.1	MÉTODO PARA A INTERPRETAÇÃO DE UM MODELO CONCEITUAL ONTOUML	72
7.1.1	Seleção dos modelos conceituais	72
7.1.2	Transcrição e identificação de padrões	73
7.1.3	Definição da heurística	74
7.2	VERIFICAÇÃO DA HEURÍSTICA	76
7.2.1	RFD derivados a partir da heurística	76
7.2.2	Verificação dos RFD gerados	79
7.3	CONSIDERAÇÕES SOBRE O CAPÍTULO	80
CAPÍTULO 8 - AMBIENTE PARA A CONSTRUÇÃO DE MODELOS CONCEITUAIS EM ONTOUML.....82		
8.1	TRABALHOS E CONCEITOS RELACIONADOS	83
8.1.1	Tipos semânticos	83
8.1.2	Desambiguação de termos	85
8.1.3	Bases semânticas.....	86
8.2	AMBIENTE COMPUTACIONAL DESENVOLVIDO.....	88
8.2.1	Funcionalidades do ambiente	89
8.2.2	Arquitetura do ambiente	90
8.3	TESTE DE VIABILIDADE	93
8.3.1	Fases do teste de viabilidade	93
8.3.2	Resultados do teste de viabilidade	95
8.3.3	Conclusões do teste de viabilidade	101
8.4	CONSIDERAÇÕES SOBRE O CAPÍTULO	102
CAPÍTULO 9 - AVALIAÇÃO DA ABORDAGEM PROPOSTA.....103		
9.1	IDENTIFICAÇÃO AUTOMÁTICA DOS ELEMENTOS DE MODELOS CONCEITUAIS EM ONTOUML	103
9.1.1	Fases da avaliação.....	104
9.1.2	Resultados da avaliação.....	105
9.1.3	Conclusões da avaliação.....	108

9.2	DERIVAÇÃO AUTOMÁTICA DOS REQUISITOS FUNCIONAIS DE DOMÍNIO A PARTIR DE MODELOS CONCEITUAIS EM ONTOUML	109
9.2.1	Fases da avaliação	109
9.2.2	Resultados da avaliação	112
9.2.3	Conclusões da avaliação	125
9.3	CONSIDERAÇÕES SOBRE O CAPÍTULO	126
CAPÍTULO 10 - CONSIDERAÇÕES FINAIS		127
10.1	RELEVÂNCIA DA PESQUISA	127
10.2	CONTRIBUIÇÕES DA PESQUISA	128
10.3	LIMITAÇÕES DA PESQUISA	130
10.4	TRABALHOS FUTUROS	130
10.5	CONCLUSÃO	131
REFERÊNCIAS DA REVISÃO SISTEMÁTICA.....		132
REFERÊNCIAS BIBLIOGRÁFICAS		136
APÊNDICE A – INSTRUMENTO PARA AVALIAR A EXPRESSIVIDADE DA ONTOUML E UML.....		145
APÊNDICE B - INSTRUMENTOS PARA AVALIAR A IDENTIFICAÇÃO AUTOMÁTICA DOS CONSTRUTOS ONTOUML		146
APÊNDICE C - INSTRUMENTOS PARA AVALIAR A DERIVAÇÃO DE REQUISITOS FUNCIONAIS DE DOMÍNIO		150
APÊNDICE D – TEXTOS UTILIZADOS NO EXPERIMENTO PARA AVALIAR A IDENTIFICAÇÃO AUTOMÁTICA DOS CONSTRUTOS ONTOUML.....		161
APÊNDICE E – RESULTADOS BRUTOS DA AVALIAÇÃO DA IDENTIFICAÇÃO AUTOMÁTICA DOS CONSTRUCTOS ONTOUML.....		163
APÊNDICE F – RESULTADOS BRUTOS DA AVALIAÇÃO DE DERIVAÇÃO DE REQUISITOS FUNCIONAIS DE DOMÍNIO		166

LISTA DE FIGURAS

Figura 1-1. Estrutura do documento da tese	10
Figura 2-1. Tipos de ontologias de acordo com o conteúdo, adaptado de (GUARINO, 1998)	19
Figura 3-1. 1ª etapa da pesquisa	22
Figura 3-2. 1ª etapa da pesquisa e resultado.....	37
Figura 4-1. Construto Universal e suas subclasses	39
Figura 4-2. Tipologia da classe Substantial Universal (GUIZZARDI, 2005).....	40
Figura 4-3. Ontologia para o domínio da Genealogia (GUIZZARDI, 2005)	42
Figura 4-4. Categorias da classe Substance Sortal	43
Figura 4-5. Aplicação de um RoleMixin (GUIZZARDI, 2005).....	44
Figura 4-6. Tipologia da classe Moment Universal	46
Figura 4-7. Representação de um Relator Universal	47
Figura 4-8. Fragmento do metamodelo UML para representação de categorias da OntoUML	48
Figura 5-1. Estrutura da pesquisa	53
Figura 6-1. 2ª etapa da pesquisa	60
Figura 6-2. Modelo conceitual em OntoUML.....	63
Figura 6-3. Modelo conceitual em UML	64
Figura 6-4. Quantidade de escolhas pelos profissionais para cada uma das doze sentenças	66
Figura 6-5. Quantidade de escolhas pelos estudantes para cada uma das doze sentenças	67
Figura 6-6. 2ª etapa da pesquisa e resultado.....	70
Figura 7-1. 3ª etapa da pesquisa.....	71
Figura 7-2. Modelo conceitual parcial OntoUML do domínio de Instituição de Saúde (ID: D)	77
Figura 7-3. Modelo conceitual parcial OntoUML do domínio de Conferência Acadêmica (ID: E).	78
Figura 7-4. 3ª etapa da pesquisa e resultado.....	81
Figura 8-1. 4ª etapa da pesquisa.....	83
Figura 8-2. Exemplo de supersense identificado pelo WordNet.	87
Figura 8-3. Principais atividades executadas por meio do ambiente computacional.....	90
Figura 8-4. Arquivo XML para a desambiguação do termo.....	92
Figura 8-5. Modelo conceitual OntoUML "ouro"	95
Figura 8-6. Passo 1: seleção do texto	96

Figura 8-7. Passo 2: resultado do topia	96
Figura 8-8. Passo 3: arquivo XML criado	99
Figura 8-9. Passo 4: desambiguação.....	99
Figura 8-10. Modelo conceitual em OntoUML gerado pelo ambiente computacional.....	100
Figura 8-11. 4ª etapa da pesquisa e resultado.....	102
Figura 9-1. 5ª etapa da pesquisa	103
Figura 9-2. Área de formação na graduação dos profissionais.....	112
Figura 9-3. Nível de formação acadêmica dos profissionais.....	113
Figura 9-4. Nível de experiência dos profissionais com levantamento de requisitos	114
Figura 9-5. Nível de experiência dos estudantes com levantamento de requisitos	115
Figura 9-6. Critérios de qualidade dos textos pelos profissionais	116
Figura 9-7. Critérios de qualidade dos textos pelos estudantes	116
Figura 9-8. Recorte Instrumento 1, lista de requisitos gerados (Comissão de Vendas)	119
Figura 9-9. Cobertura do método visão geral pelos profissionais	121
Figura 9-10. Cobertura do método visão geral pelos estudantes	121
Figura 9-11. Percepções dos profissionais	123
Figura 9-12. Percepções dos estudantes.....	123
Figura 9-13. 5ª etapa da pesquisa e resultado.....	126
Figura 10-1. Etapas da pesquisa e resultados	128
Figura 10-2. Avaliação dos termos pelos especialistas, instrumento 1	163
Figura 10-3. Avaliação dos termos pelos especialistas, instrumento 2.....	164
Figura 10-4. Indicação de termos relevantes não identificados pelo método	165
Figura 10-5. Observações dos profissionais parte 1	166
Figura 10-6. Observações dos profissionais parte 2.....	167
Figura 10-7. Observações dos estudantes	168

LISTA DE TABELAS

Tabela 3-1. Critérios de exclusão.....	25
Tabela 3-2. Atividades da ERS onde as ontologias são aplicadas	27
Tabela 3-3. Função das ontologias	29
Tabela 3-4. Classe e linguagens usadas pelas ontologias	30
Tabela 3-5. Domínio do conhecimento representado pelas ontologias	31
Tabela 3-6. Domínio do conhecimento da ERS representado pelas ontologias	32
Tabela 3-7. Tipo de avaliação aplicada nos estudos	33
Tabela 3-8. Contexto da avaliação empírica aplicada nos estudos	34
Tabela 3-9. Contribuição da ontologia aplicada nos estudos.....	34
Tabela 6-1. Resultado consolidado do grupo de profissionais.....	65
Tabela 6-2. Resultado consolidado do grupo de estudantes	66
Tabela 6-3. Escolhas dos estudantes por classe	67
Tabela 7-1. Quantidade de elementos do modelo conceitual por domínio	72
Tabela 7-2. Variáveis para a verificação da consistência da heurística.....	79
Tabela 7-3. Quantidade de elementos sem RFD listados por domínio.....	79
Tabela 9-1. Dados sumarizados dos textos utilizados no experimento	104
Tabela 9-2. Perfil dos especialistas em OntoUML participantes do experimento	105
Tabela 9-3. Resultado das medidas coletadas para cada texto	106
Tabela 9-4. Resultado das métricas para cada texto	106
Tabela 9-5. Anos de atuação dos profissionais com levantamento de requisitos.....	113
Tabela 9-6. Anos de atuação dos estudantes com levantamento de requisitos	114
Tabela 9-7. Critérios de qualidade por texto e por grupo de participantes	117
Tabela 9-8. Percentual de requisitos incorretos gerados pelo método por texto	119
Tabela 9-9. Cobertura do método por texto na visão de profissionais e estudantes	122

LISTA DE QUADROS

Quadro 3-1. Questões de pesquisa da revisão sistemática	23
Quadro 3-2. String de busca	24
Quadro 3-3. Categorias, subcategorias e critérios para a classificação dos resultados	26
Quadro 4-1. Estereótipos da OntoUML referentes aos Substantial Universals	44
Quadro 4-2. Estereótipos da OntoUML referentes aos Moments Universals	49
Quadro 5-1. Medidas coletadas no experimento	58
Quadro 5-2. Métricas utilizadas no experimento	58
Quadro 6-1. Descrição do domínio	61
Quadro 6-2. Seleção da linguagem mais expressiva por sentença e por grupo	68
Quadro 7-1. Algoritmo parcial para a leitura e extração dos RFD a partir de modelos conceituais em OntoUML.	75
Quadro 7-2. RFD parciais relacionados ao domínio de Instituição de Saúde (ID: D).	77
Quadro 7-3. RFD parciais relacionados ao domínio de Conferência Acadêmica (ID: E).....	78
Quadro 8-1. Mapeamento dos tipos semânticos associados a substantivos (CASTRO, 2010)	85
Quadro 8-2. Agrupamento léxico do WordNet	87
Quadro 8-3. Mapeamento direto entre os supersenses e os tipos semânticos	88
Quadro 8-4. Texto selecionado (Gemino e Wand, 2005).....	94
Quadro 8-5. Termos “ouro” do modelo ER (Gemino and Wand, 2005).	94
Quadro 8-6. Termos relevantes extraídos pelo Topia versus termos ouro	96
Quadro 8-7. Termos ouro versus termos relevantes topia.	97

LISTA DE ABREVIATURAS E SIGLAS

AIML	Artificial Intelligence Markup Language
AORML	<i>Agent-Object Relationship Modeling Language</i>
BWW	<i>Bunge Wand e Weber</i>
ER	<i>Entidade Relacionamento</i>
ERS	Engenharia de Requisitos de Software
FCA	<i>Formal Concept Analysis</i>
FOL	First-Order Logic
KIF	<i>Knowledge Interchange Format</i>
LEL	Léxico Estendido da Linguagem
OIL	<i>Ontology Inference Layer or Ontology Interchange Language</i>
OWL	<i>Web Ontology Language</i>
PLN	Processamento de Linguagem Natural
TF-IDF	<i>Term Frequency - Inverse Document Frequency</i>
UFO	<i>Unified Foundational Ontology</i>
UML	<i>Unified Modeling Language</i>
WER	Workshop em Engenharia de Requisitos
XML	<i>eXtensible Markup Language</i>
WSD	<i>Word Sense Disambiguation</i>

CAPÍTULO 1 - INTRODUÇÃO

They did not know it was impossible, so they did it!

Mark Twain.

A Engenharia de Requisitos de Software (ERS) é um processo sistemático de desenvolvimento de requisitos que inclui a análise do problema, a documentação dos resultados observados, o uso de diversas formas de representação e a verificação da acurácia do entendimento obtido. Todo este processo ocorre de maneira interativa e cooperativa (LOUCOPOULOS; KARAKOSTAS, 1995). Além disso, a ERS fornece os mecanismos apropriados para entender aquilo que o cliente deseja, analisando as necessidades, avaliando a viabilidade, negociando uma solução razoável, especificando uma solução sem ambiguidades, validando a especificação e gerenciando as necessidades à medida que são transformadas em um sistema (THAYER; DORFMAN, 1997).

Neste processo são identificadas quatro atividades principais: (i) Elicitação, uso de técnicas para ajudar a explicitar as principais necessidades dos stakeholders, as quais serão transformadas em requisitos do software; (ii) Negociação, os stakeholders definem as prioridades dos requisitos e decidem quais requisitos serão implementados no software; (iii) Especificação/Documentação, os requisitos podem ser especificados e documentados usando linguagem natural e/ou modelos formais; e (iv) Verificação/Validação de requisitos, é confrontado se o que foi documentado corresponde às necessidades originais dos usuários (POHL, 1997).

A elicitação de requisitos é a primeira atividade neste processo, e os problemas relacionados a esta atividade são os que mais afetam o sucesso dos projetos de desenvolvimento de software (BOEHM, 1981)(GAO, 1992). Os requisitos elicitados geralmente são incompletos, incompreensíveis e ambíguos (MARTINS; DALTRINI, 1999). A falta de habilidade para discutir problemas tem sido uma das maiores deficiências da teoria e da prática em software (ZANLORENCI; BURNETT, 1998). Esta deficiência impacta principalmente na tarefa de identificar as reais necessidades dos usuários.

A elicitação de requisitos é um processo que envolve diversos métodos e técnicas, os quais infelizmente ainda não satisfazem as exigências da indústria, apesar dos evidentes avanços (MACEDO; LEITE, 1999). A aquisição do conhecimento de um determinado domínio é um processo caro (GUIZZARDI, 2005), pois necessidades mal compreendidas podem afetar negativamente todas as etapas posteriores do desenvolvimento do software. Conseqüentemente, o processo de ERS é complexo e difícil de gerenciar, pois envolve o tratamento de pontos de vistas diversos entre os envolvidos que buscam uma compreensão compartilhada e de consenso do sistema a ser desenvolvido (ROBINSON; PAWLOWSKI, 1999).

Nas fases iniciais da ERS, nas quais ocorrem as primeiras interações com os *stakeholders*, são comuns os seguintes problemas: as informações sobre o sistema a ser desenvolvido são variavelmente imprecisas e inconsistentes (JURETA et al., 2010); a falta de entendimento do negócio pelos engenheiros de requisitos (OLIVEIRA et al., 2004) e a deficiência nas comunicações entre os especialistas no negócio e da computação comprometem a qualidade das informações (LUIS et al., 2008); e a falta de consenso sobre o uso dos termos na organização pode levar a significados distintos (GARRIDO et al., 2007).

Considerando este contexto, algumas das principais necessidades durante o desenvolvimento de software, especialmente em suas fases iniciais são: o uso de uma linguagem comum que permita o entendimento compartilhado entre os *stakeholders* e, assim, promova a coesão entre as informações obtidas de diversas origens (LEE; GANDHI, 2005); uma compreensão abrangente do domínio (LEE; GANDHI, 2005); um modelo conceitual que envolva um vocabulário comum utilizado entre os distintos *stakeholders* para facilitar a discussão dos elementos que podem aparecer no sistema; e a construção de uma especificação suficientemente clara, sem ambiguidades e traduzível para outras formas de representação. Na tentativa de atender ou minimizar estas e outras necessidades decorrentes dos problemas relacionados à ERS, as ontologias têm sido um importante recurso.

Há diferentes definições sobre ontologias, que são geradas por pontos de vistas diferentes e complementares a uma mesma realidade (CORCHO et al., 2003). Para Gruber (GRUBER, 1993A), por exemplo, ontologia é uma especificação compartilhada de uma conceituação. Já Guarino (GUARINO, 1998) discute a definição de Gruber incluindo o aspecto intencional para esclarecer a diferença entre uma ontologia e uma conceituação e estabelece uma visão de construir ontologias

usando teoria lógica. Para Guarino, uma ontologia é considerada um conjunto de axiomas lógicos que corresponde ao significado intencional de um vocabulário.

De uma maneira geral, as ontologias têm sido aplicadas na ERS principalmente com os seguintes objetivos: compartilhamento de um vocabulário comum (ARANDA et al., 2008); reuso e estruturação do conhecimento (GIRARDI; LEITE, 2008); entendimento do domínio do problema (LI, 2005); análise da expressividade da linguagem (HARZALLAH et al., 2012); representação do domínio mais próxima do mundo real (ZHANG et al., 2007); e melhoria da comunicação entre especialistas de diversos domínios (KILOV; SACK, 2009).

As possibilidades de aplicação das ontologias na ERS são diversas, porém tem-se observado um papel importante das ontologias nas linguagens de modelagem conceitual. Mylopoulos (MYLOPOULOS, 1992) define a modelagem conceitual como a atividade de descrever formalmente aspectos do mundo físico e social com o propósito de entendimento. Nesta abordagem, a modelagem trata da modelagem da realidade e não da modelagem do sistema computacional (GUIZZARDI, 2005) (GUIZZARDI et al., 2012)

Diversos trabalhos têm sido propostos usando o conceito de ontologia para avaliar a expressividade de linguagens de modelagem com o objetivo de identificar deficiências na gramática utilizada (WAND et al., 1995), (ROSEMANN, GREEN, 2002), (WAGNER, 2003), (GEMINO; WAND, 2005), (CAÑETE-VALDEÓN et al., 2009), (BERA; EVERMANN, 2012). As propostas mostram a importância em se utilizar técnicas e linguagens de modelagem ontologicamente fundamentadas. Os conceitos de um universo de discurso são entidades abstratas que existem apenas na mente dos usuários e para que eles sejam capturados, eles precisam ser representados em um artefato concreto, ou seja, uma especificação. Isto implica o uso de uma linguagem para representá-los de maneira concisa, completa e não ambígua. Uma especificação produzida deve ser uma representação compartilhada de entidades de um domínio que os especialistas consideram relevante em um universo de discurso, o qual pode ser utilizado para promover a solução de problemas, comunicação, aprendizagem e reuso em alto nível de abstração (ARANGO, 1994).

O uso de uma linguagem com falhas de expressividade pode comprometer o entendimento dos artefatos de requisitos em fases posteriores. Segundo Mylopoulos (MYLOPOULOS, 1992), a adequação de uma notação de modelagem conceitual

está baseada na sua contribuição para a construção de modelos que representam a realidade e promovam um entendimento comum entre seus usuários. Henderson-Sellers et al. (HENDERSON-SELLERS et al., 2015) reforçam a importância da notação da linguagem ao afirmar que bons modelos conceituais precisam de boas linguagens de modelagem. Nesta perspectiva, Guizzardi (GUIZZARDI, 2005) enfatiza a necessidade do uso de linguagens com primitivas ontologicamente bem fundamentadas que ajudem a representar o máximo possível a realidade do domínio de um problema.

Uma vez que os modelos conceituais ontologicamente bem fundamentados proveem uma representação mais próxima da realidade de um domínio, alguns dos problemas recorrentes da atividade de requisitos podem ser minimizados com a intensificação do seu uso. Segundo Falbo et al. (FALBO et al., 1998) um modelo de domínio pode ser usado na aplicação da análise de requisitos para melhorar a comunicação e o entendimento do domínio e ajudar o processo de elicitação de requisitos. Desta maneira, no escopo desta pesquisa, considerou-se que modelos conceituais ontologicamente bem fundamentados podem ser um instrumento importante a ser utilizado na atividade de elicitação de requisitos.

1.1 Motivação

Motivado pela necessidade de uma linguagem fundamentada ontologicamente que provesse a semântica necessária para a construção de modelos conceituais e que representasse conceitos mais próximos à realidade, Guizzardi (GUIZZARDI, 2005) propôs a linguagem OntoUML. O principal objetivo da OntoUML é atingir o comprometimento ontológico falho em outras linguagens genéricas para modelagem conceitual. Para alcançar este objetivo, a OntoUML foi baseada em teorias da metafísica, ciências cognitivas e linguísticas.

Em sua proposta, Guizzardi discute problemas clássicos de modelagem conceitual provenientes da falta de adequação da linguagem utilizada. As classes propostas na OntoUML são especializações das classes abstratas da *Unified Foundational Ontology* (UFO) e estendem o metamodelo original da *Unified Modeling Language* (UML). Em trabalhos como este, a ontologia de fundamentação tem o objetivo de prover semântica para linguagens genéricas de modelagem conceitual e, assim, restringir as possíveis interpretações de suas primitivas de modelagem (GUIZZARDI, 2005).

Na área de modelagem conceitual, outras linguagens e metamodelos são mais populares nas empresas do que a OntoUML, como por exemplo, o meta modelo ER (Entidade Relacionamento) e a UML. Apesar da facilidade de uso destas linguagens, elas sofrem de problemas de expressividade, tais como: sobrecarga, redundância, falta de clareza na linguagem UML e em outras linguagens de modelagem (Guizzardi, 2005).

A OntoUML possibilita maior expressividade nos modelos conceituais, evitando a sobrecarga e a redundância (GUIZZARDI, 2005). Neste sentido, a OntoUML tem sido aplicada com sucesso na captura de conhecimento em diversos domínios tais como Eletro física (GONÇALVES et al., 2007), Telecomunicações (BARCELOS et al., 2011) e Petróleo e Gás (GUIZZARDI et al., 2010). No entanto, a sua utilização é mais complexa do que linguagens tradicionais, como por exemplo, a UML ou o ER, principalmente para os modeladores iniciantes (GUIZZARDI et al., 2011). Uma das dificuldades na construção de um modelo representado em OntoUML é a identificação do construto correto para um determinado conceito a ser representado. Ainda há pouco suporte metodológico para ajudar o usuário a decidir como representar os elementos que denotam propriedades universais em um dado domínio, e, muitas vezes, escolhas de modelagem são frequentemente feitas de maneira *ad hoc* (GUIZZARDI, 2005).

Guizzardi (GUIZZARDI, 2005) ressalta a necessidade de métodos que ajudem o usuário a decidir como modelar os elementos de uma conceitualização usando uma linguagem de modelagem na qual os princípios ontológicos estão incorporados na sintaxe. Também destaca a importância de métodos que apoiem a aplicação de padrões ontológicos de projeto para resolver problemas clássicos e recorrentes da modelagem conceitual. Desta maneira, há lacunas a serem preenchidas no que diz respeito ao desenvolvimento de mecanismos automáticos ou semiautomáticos que apoiem o modelador do domínio na identificação destes conceitos e seu construto correto.

No sentido de preencher estas lacunas, Castro (CASTRO, 2010) propõe uma abordagem linguística para a modelagem conceitual utilizando como linguagem a OntoUML e a teoria de Dixon (DIXON, 2005) para identificação dos tipos semânticos. Após definir manualmente o mapeamento entre os tipos semânticos de Dixon e os construtos OntoUML, Castro (CASTRO, 2010) avalia a ferramenta PALAVRAS como uma alternativa para a análise semântica automática. No entanto,

ele concluiu que a ferramenta não fornece nenhum tipo de classificação para os significados dos verbos, que são o centro das orações, e, mesmo para os substantivos, as anotações estão sujeitas a falhas e a erros de identificação. Leão et al. (LEÃO et al., 2012) também realizaram a mesma análise com outras ferramentas e chegaram à conclusões semelhantes a Castro (CASTRO, 2010).

Leão et al. (LEÃO et al., 2013) estenderam a proposta de Castro (CASTRO, 2010) e propuseram um método semiautomático para a construção de ontologias bem fundamentadas por meio da desambiguação de termos. Na proposta de Leão et al. são utilizadas técnicas de Processamento de Linguagem Natural (PLN) para a identificação de termos. Posteriormente, estes termos são desambiguados utilizando uma base semântica e, por fim, um mapeamento dos tipos semânticos aos construtos da OntoUML é realizado. No entanto, a desambiguação dos termos para a identificação correta do tipo semântico está sujeita à qualidade da base semântica utilizada.

No contexto da Engenharia de Requisitos de Software, um dos principais benefícios do modelo conceitual ontologicamente fundamentado, é a promoção do entendimento de um domínio e a melhoria da comunicação. Neste sentido, considera-se o modelo conceitual representado em OntoUML um importante instrumento para a eliciação de requisitos de software, pois pode ser aplicado para a compreensão de um domínio de conhecimento, principalmente os domínios mais complexos. Espera-se que um modelo conceitual ontologicamente fundamentado seja um instrumento facilitador na derivação dos requisitos de software.

No entanto, há a necessidade de se pesquisar, avaliar e propor métodos automáticos e semiautomáticos que apoiem os envolvidos no processo de eliciação de requisitos de software a capturar os conceitos relevantes de um domínio e, por meio deles, construir o modelo conceitual ontologicamente fundamentado. Foram identificadas iniciativas neste sentido, no entanto, também foram identificadas oportunidades de contribuição para o preenchimento das seguintes lacunas: avaliação de algoritmos para extração de termos usando técnicas de PLN; ferramenta para apoiar a construção semiautomática do modelo conceitual ontologicamente fundamentado; e método para apoiar a derivação automática dos requisitos de software a partir do modelo conceitual. As ferramentas e os métodos propostos devem estar integrados e coordenados de maneira a apoiar os envolvidos em todas as etapas do processo de derivação de requisitos funcionais do domínio.

1.2 Objetivos

Considerando os problemas expostos anteriormente, a questão que esta pesquisa pretende responder é: **como modelos conceituais em OntoUML podem apoiar a derivação de requisitos funcionais a partir de descrições de domínio?** E para responder esta questão, o objetivo geral desta tese é **propor uma abordagem centrada em modelos conceituais em OntoUML para apoiar a derivação de requisitos funcionais a partir de descrições de domínio.**

Para atender o objetivo geral, os seguintes objetivos específicos devem ser atingidos:

- i. Conceber um método para apoiar a construção de modelos conceituais em OntoUML a partir de descrições de domínio.
- ii. Conceber um método para apoiar a derivação de requisitos funcionais de domínio a partir de modelos conceituais em OntoUML.
- iii. Desenvolver apoio computacional para a construção de modelos conceituais em OntoUML.
- iv. Desenvolver apoio computacional para a derivação de requisitos funcionais de domínio a partir de modelos conceituais em OntoUML.
- v. Avaliar a abordagem proposta.

1.3 Delimitação de escopo

Para que os resultados desta tese sejam bem compreendidos e que expectativas não sejam frustradas, seguem os principais pontos que delimitam o seu escopo.

A modelagem conceitual, compreendida nesta proposta, trata da representação de modelos da realidade, e não de modelos computacionais. Embora, o modelo conceitual possa ser um importante ponto de partida para a derivação dos modelos computacionais, os quais são desenvolvidos na etapa de *design* e implementação, não é o objetivo deste trabalho avançar nestes modelos. Também não foi previsto no escopo desta pesquisa o gerenciamento de mudanças de requisitos.

Uma das motivações desta tese foi o uso de modelo conceitual ontologicamente fundamentado como um instrumento para a elicitação de requisitos. Porém, a expectativa foi extrair apenas requisitos de alto nível. Para melhor detalhamento destes requisitos e aprofundamento das regras de negócio, será

necessário o uso combinado de outras técnicas de elicitação, como, por exemplo, entrevistas, questionários e avaliação de documentação. Devido à natureza do modelo conceitual, o objetivo foi extrair apenas os requisitos funcionais. Os requisitos não funcionais não estão no escopo desta tese.

No que se refere à proposta de apoio computacional para a construção do modelo conceitual, não era esperado a construção completa de modo automático, mas sim, um processo apoiado por métodos e ferramentas integradas que auxiliasse nas principais atividades, desde a construção do modelo conceitual até a derivação dos requisitos funcionais de domínio.

Também é importante pontuar, que na área de PLN as ferramentas disponíveis para o idioma português não atendem as funcionalidades necessárias para a implementação do método proposto para a construção do modelo conceitual em OntoUML, principalmente para a desambiguação dos termos. Portanto, todas as descrições de domínio utilizados em experimentos foram em inglês devido à disponibilidade de ferramentas.

A elicitação de requisitos é uma atividade que depende da experiência e do conhecimento do profissional a respeito de um determinado domínio. Desta maneira, a abordagem proposta nesta tese deve ser um suporte para profissionais com menos conhecimento no domínio do problema em análise, ou seja, a abordagem é um apoio, principalmente, para a utilização por profissionais iniciantes. Por fim, uma das maiores motivações foi o uso da abordagem proposta em domínios mais complexos, nos quais o conhecimento do negócio é fundamental para uma elicitação de requisitos mais precisa. Considera-se domínios mais complexos aqueles cujo negócio não é amplamente conhecido pelo público.

1.4 Estrutura do documento da tese

Para melhor entendimento de como a tese e os resultados são apresentados neste documento, a Figura 1-1 representa a estrutura principal do documento da tese.

O Capítulo 1, aqui apresentado, visa oferecer ao leitor um panorama geral sobre o contexto no qual se insere este trabalho de pesquisa, além de estabelecer a motivação, objetivo geral, específico e as questões de pesquisa.

Os Capítulos 2, 3, 4 aprofundam o referencial teórico. O Capítulo 2 aborda os temas Engenharia de Requisitos e Ontologias. Posteriormente, no Capítulo 3 é

apresentada uma revisão sistemática sobre a aplicação de ontologias na área de Engenharia de Requisitos e o Capítulo 4 aprofunda o referencial teórico sobre OntoUML.

O Capítulo 5 apresenta a estrutura da pesquisa, indicando etapas, métodos e as estratégias de execução.

Os Capítulos 6, 7, 8 e 9 estão focados na apresentação dos resultados da tese. O Capítulo 6 apresenta o resultado de um experimento que avaliou a expressividade da OntoUML. O Capítulo 7 apresenta o método proposto para derivar requisitos funcionais a partir de modelos conceituais em OntoUML. O Capítulo 8 apresenta o ambiente construído para apoiar a abordagem proposta. E, finalmente, o Capítulo 9 apresenta o resultado de duas avaliações. A primeira com o objetivo de avaliar o método para a construção semiautomática do modelo conceitual em OntoUML e a segunda com o objetivo de avaliar o método para a derivação dos requisitos funcionais de domínio.

O Capítulo 10 é o fechamento desta tese com a apresentação da relevância, contribuições, limitações e trabalhos futuros da pesquisa.

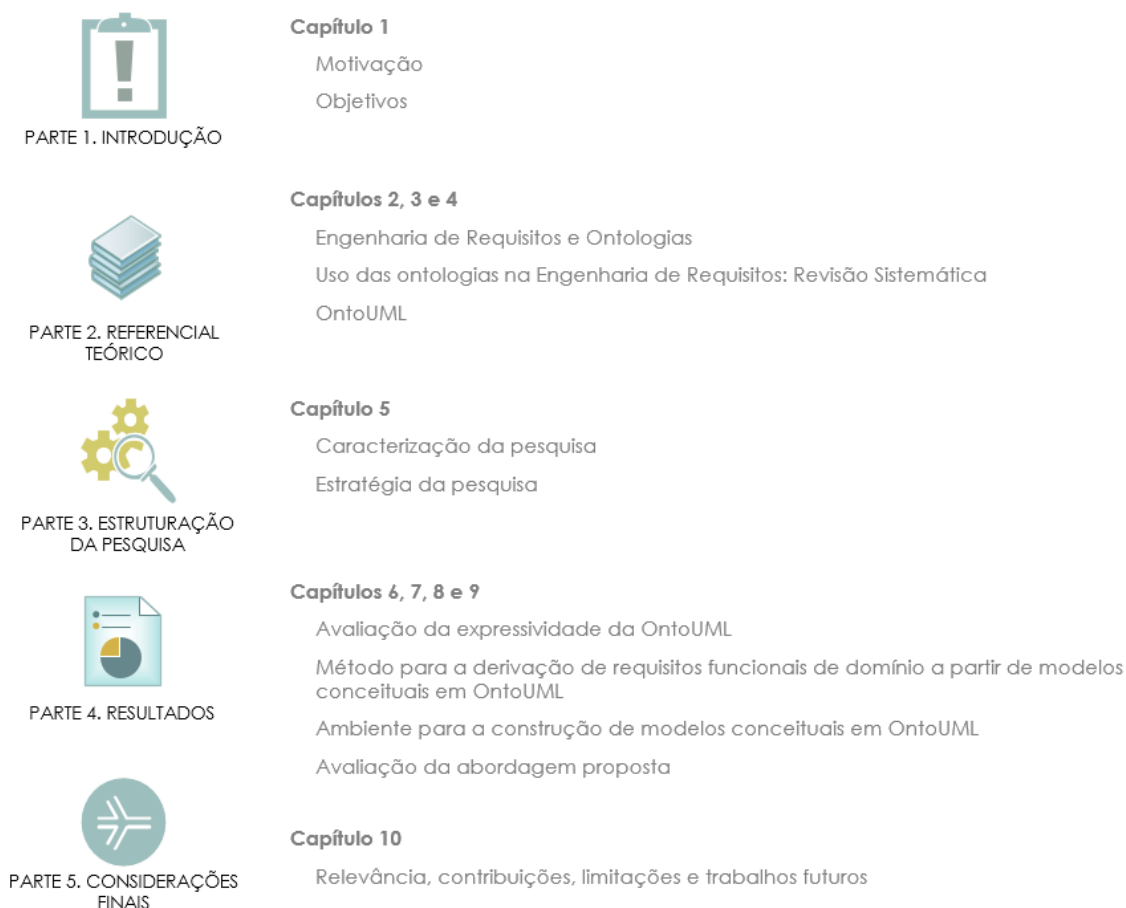


Figura 1-1. Estrutura do documento da tese

1.5 Considerações sobre o capítulo

Este capítulo apresentou a elicitação de requisitos como uma das atividades mais importantes e impactantes no processo de desenvolvimento de software. Apesar dos evidentes esforços ainda há diversos problemas a serem resolvidos. As ontologias são discutidas como sendo um importante recurso a ser aplicado na solução de parte destes problemas.

Neste contexto, o modelo conceitual ontológico, representado na linguagem OntoUML, é apresentado como um importante instrumento para apoiar a atividade de elicitação de requisitos. No entanto, a sua construção não é uma tarefa fácil, principalmente para os profissionais iniciantes. Neste sentido são apresentadas algumas lacunas e a proposta de uma abordagem que apoie a construção de um modelo conceitual ontológico para ser utilizado como um instrumento de derivação de requisitos funcionais de domínio. Por fim, foram apresentados os objetivos da pesquisa, a delimitação de escopo e o processo de trabalho.

CAPÍTULO 2 - ENGENHARIA DE REQUISITOS E ONTOLOGIAS

Neste capítulo são introduzidos os dois principais conceitos aplicados na execução da pesquisa. Inicialmente discute-se a Engenharia de Requisitos de Software, com destaque à elicitação de requisitos, atividade principal de interesse nesta pesquisa. Na sequência são apresentadas as ontologias, formalismo cuja contribuição para a área de Engenharia de Requisitos tem sido bastante significativa.

2.1 Engenharia de requisitos de software

O processo de desenvolvimento de uma especificação de requisitos de software tem sido chamado de Engenharia de Requisitos de Software (ERS). A correta compreensão (elicitação), a documentação (especificação) e a validação dos requisitos de software têm sido cruciais para a qualidade do software.

Não há uma definição comum para o conceito de ERS, existindo diversas definições de acordo com o foco (POHL, 1997). Segundo Loucopoulos e Karakostas (LOUCOPOULOS; KARAKOSTAS, 1995), a ERS é um processo sistemático de desenvolvimento de requisitos que inclui a análise do problema, a documentação dos resultados observados, o uso de diversas formas de representação e a verificação da acurácia do entendimento obtido. Todo este processo ocorre de maneira interativa e cooperativa.

A ERS fornece os mecanismos apropriados para entender aquilo que o cliente deseja, analisando as necessidades, avaliando a viabilidade, negociando uma solução razoável, especificando uma solução sem ambiguidades, validando a especificação e gerenciando as necessidades à medida que são transformadas em um sistema (THAYER; DORFMAN, 1997).

Castro (CASTRO, 1995) sugere quatro atividades principais que envolvem o processo de Engenharia de Requisitos: (i) Elicitação dos Requisitos, na qual se procura levantar os requisitos existentes e pretendidos pela organização, sejam eles funcionais e não-funcionais; (ii) Análise e Negociação dos Requisitos, nos quais se

faz a análise sobre os requisitos levantados na fase anterior, descobrindo-se possíveis conflitos e necessidades de elicitar outros requisitos, fatos estes negociados com todas as partes envolvidas; (iii) Documentação dos Requisitos, por meio de um documento que sirva como base para o desenvolvimento do sistema requerido; e (iv) Validação dos Requisitos, na qual serão levantadas as possíveis inconsistências e determinada a completude, ou não, dos requisitos levantados.

De maneira semelhante, Pohl (POHL, 1997) descreve as seguintes atividades principais no processo de Engenharia de Requisitos: (i) Elicitação, uso de técnicas para ajudar a explicitar as principais necessidades dos *stakeholders*, as quais serão transformadas em requisitos do software; (ii) Negociação, os *stakeholders* definem as prioridades dos requisitos e decidem quais requisitos serão implementados no software; (iii) Especificação/Documentação, os requisitos podem ser especificados e documentados usando linguagem natural e/ou modelos formais e; (iv) Verificação/Validação de requisitos, é confrontado se o que foi documentado corresponde às necessidades originais dos usuários.

Conforme estudo conduzido por Valaski et al. (VALASKI et al., 2013a), (VALASKI et al., 2013b), usando como base os 258 artigos publicados no Workshop em Engenharia de Requisitos (WER), foi identificado que os temas elicitação de requisitos e modelagem de requisitos são os mais discutidos na ERS ao longo de suas 15 edições. O estudo deixa evidente que ainda são necessários esforços na solução de problemas relacionados a estes temas.

2.1.1 Elicitação de requisitos

Os problemas relacionados à atividade de elicitação de requisitos são os que mais afetam o sucesso dos projetos de desenvolvimento de software (BOEHM, 1981); (GAO, 1992). Os requisitos elicitados geralmente são incompletos, incompreensíveis e ambíguos (MARTINS; DALTRINI, 1999). A falta de habilidade para discutir problemas tem sido uma das maiores deficiências da teoria e da prática em software (ZANLORENCI; BURNETT, 1998). Esta deficiência impacta principalmente na tarefa de identificar as reais necessidades dos usuários.

A elicitação de requisitos é um processo que envolve diversos métodos e técnicas, os quais infelizmente ainda não satisfazem as exigências da indústria, apesar dos evidentes avanços que têm acontecido nos últimos anos (MACEDO; LEITE, 1999). A deficiência na elicitação de requisitos ocorre pelo fato de que,

persiste na área de Engenharia de Software, a tentativa de visualizar a tecnologia como solução de problema, sem antes focar no entendimento do problema e na negociação de eventuais conflitos de interesses pela solução (ZANLORENCI; BURNETT, 1998). Diversas técnicas têm sido propostas com o objetivo de melhorar o processo de elicitación de requisitos, algumas destas técnicas são discutidas a seguir.

2.1.2 Técnicas de elicitación de requisitos

O objetivo das técnicas de elicitación de requisitos é auxiliar o Engenheiro de Requisitos a identificar o conhecimento e os requisitos dos stakeholders (POHL; RUPP, 2011). Não existe um método único para elicitación de requisitos, sendo necessário considerar as técnicas compatíveis e mais apropriadas ao cenário encontrado (HICKEY; DAVIS, 2003).

A aplicação das técnicas, levando em consideração as restrições culturais, organizacionais ou do domínio, permite que os requisitos sejam elicitados de forma mais ampla e completa possível. Técnicas diferentes podem ser combinadas de maneira a potencializar os pontos fortes de cada técnica. De uma maneira geral, os tipos de técnicas de elicitación são classificadas em: técnicas de pesquisa, técnicas de observação, técnicas de criatividade, técnicas baseadas em documentos e técnicas de apoio.

2.1.2.1 Pesquisa

As técnicas de pesquisa têm o objetivo de estimular as declarações dos *stakeholders* a respeito de seus requisitos, uma vez que é pressuposto que o respondente é capaz de explicitar o conhecimento necessário. São exemplos desta técnica, a entrevista e o questionário.

2.1.2.2 Observação

Em situações em que os especialistas do domínio não dispõem de tempo necessário para compartilhar seus conhecimentos, ou ainda, se não são capazes de expressar com segurança seus conhecimentos, as técnicas de observação podem ser mais apropriadas. A observação pode ser: em campo, quando o engenheiro de requisitos apenas observa os procedimentos a serem realizados pelo especialista do

domínio; ou *apprenticing*, quando o engenheiro de requisitos precisa participar ativamente da execução dos procedimentos a serem observados.

2.1.2.3 *Criatividade*

As técnicas de criatividade têm o objetivo de desenvolver requisitos inovadores, mas, geralmente, não são muito adequadas para estabelecer requisitos precisos sobre o comportamento do sistema. São exemplos desta técnica, o *brainstorming* e a técnica de analogias.

2.1.2.4 *Documentos*

As técnicas baseadas em documentos geralmente reutilizam soluções já implementadas em sistemas existentes, principalmente quando se trata de sistemas legados. Documentos que regem um domínio, como, por exemplo, domínio legal, podem ser utilizados como fonte para eliciação de requisitos.

2.1.2.5 *Apoio*

As técnicas de apoio são instrumentos adicionais que podem ser usados em conjunto com as demais técnicas. Exemplos destes apoios são: os mapas mentais, os workshops, as gravações de áudio e vídeo e os protótipos.

Além das técnicas já mencionadas, que são mais tradicionais e mais populares, existem também instrumentos para eliciação de requisitos de uso mais comum no meio científico. Exemplos destas técnicas são: o Léxico Estendido da Linguagem (LEL) (FRANCO; LEITE, 1992), a Linguagem i* (iStar) (YU, 1995) e os Cenários (JACOBSON, 1992).

2.1.2.6 *LEL*

O LEL é um metamodelo projetado para capturar o vocabulário da aplicação e usa linguagem natural para representação de seus símbolos. O LEL é composto por um conjunto de símbolos que representam a linguagem da aplicação. Os símbolos são, em geral, palavras ou frases que o cliente repete ou enfatiza. Enquanto se obtêm os símbolos, os engenheiros de software compreendem os seus significados. Os símbolos obtidos se classificam em sujeito, objeto, verbo e estado. O LEL tem o objetivo de entender a linguagem do problema sem se preocupar com o problema em si (LEITE et al., 1997).

2.1.2.7 iStar

O iStar é um framework orientado a agentes e desenvolvido para modelar relações de intenções entre atores estratégicos. Seu modelo é usado para a descrição de dependências sociais entre os atores do sistema. Permite descrever aspectos de intencionalidade e motivações envolvendo atores em um ambiente organizacional. Para descrever estes aspectos são propostos dois modelos: o Modelo de Dependências Estratégicas (SD) e o Modelo de Razões Estratégicas (SR). Seus principais elementos são os atores e suas dependências. Alguns dos seus elementos principais são: *depend*, *depende*, *dependum*, *goal*, *softgoal*, *task* e *resource* (YU, 1995).

Embora o iStar, não seja uma técnica de elicitação de requisitos, ele é um framework apoia o entendimento estratégico da organização o que indiretamente contribuiu para melhor identificação dos requisitos de um sistema.

2.1.2.8 Cenários

A técnica de cenários tem o objetivo de representar uma visão geral do sistema. Os cenários, normalmente chamados de casos de uso, fornecem uma descrição de como o sistema será utilizado (JACOBSON, 1992). Nesta técnica são identificados os roteiros de uso para o sistema a ser construído. Os principais elementos nesta técnica são os atores, os casos de usos e as relações entre estes. Segundo Santander e Castro (SANTANDER; CASTRO, 2000), os modelos gerados pela técnica i* podem auxiliar o desenvolvimento de cenários sob a forma de casos de uso. Desta maneira, artefatos gerados na primeira técnica podem ser integrados e reutilizados na aplicação da segunda técnica.

Embora existam técnicas e métodos que apoiem o processo de elicitação de requisitos, o entendimento dos requisitos de software ainda é um desafio. A elicitação de requisitos ocorre nas fases iniciais da ERS, nas quais se estabelecem as primeiras interações com os *stakeholders*. Nesta fase, as informações sobre o sistema a ser desenvolvido são variavelmente imprecisas e inconsistentes (JURETA et al., 2010). Parte destes problemas é decorrente da falta de entendimento do negócio pelos engenheiros de requisitos (OLIVEIRA et al., 2004), da falta de consenso sobre o uso dos termos na organização (GARRIDO et al., 2007) e da deficiência nas comunicações entre os especialistas de negócio e de computação (LUIIS et al., 2008).

Considerando estas questões, pode-se citar alguns exemplos de necessidades à elicitación de requisitos: o uso de uma linguagem comum que permita o entendimento compartilhado entre os stakeholders e, assim, promover coesão entre as informações obtidas de diversas origens (LEE; GANDHI, 2005); uma compreensão abrangente do domínio de negócio (LEE; GANDHI, 2005); um modelo conceitual que envolva um vocabulário comum utilizado entre os distintos stakeholders para facilitar a discussão dos elementos que podem aparecer no sistema; e a construção de uma especificação suficientemente clara, sem ambiguidades e traduzível para outras formas de representação. As ontologias têm sido um importante recurso utilizado para o atendimento destas e outras necessidades decorrentes dos problemas relacionados à ERS. Em (VALASKI et al., 2013c) é ressaltada a necessidade de apoio semântico à ERS, principalmente no que se refere à atividade de elicitación de requisitos, por meio de ontologias.

2.2 Ontologias

Ontologia é um tipo de formalismo utilizado para a representação do conhecimento e tem sido utilizada principalmente com os seguintes objetivos: prover uma compreensão comum compartilhada de uma estrutura de informação entre pessoas ou agentes de software; possibilitar o reúso de domínios de conhecimento; fazer suposições explícitas de um domínio e separar o domínio de conhecimento do domínio operacional (NOY; MCGUINNESS, 2001).

O termo ontologia pode significar coisas distintas para membros de comunidades diferentes e, historicamente, tem origem na Filosofia. Na comunidade da Ciência da Computação, o termo passou a ser utilizado no contexto de compartilhamento da informação referente a descrições formais de domínios particulares. Na comunidade de Inteligência Artificial, o termo se popularizou com o uso das ontologias para compartilhamento de conhecimento e reúso (LACY, 2005).

Ontologias têm sido utilizadas para propósitos diferentes (processamento de linguagem natural, gerenciamento do conhecimento, e-commerce, integração inteligente de informações, web semântica e outros) e em comunidades diferentes (Engenharia do Conhecimento, Banco de Dados, Engenharia de Software e outros) (CORCHO et al., 2003). Na Web Semântica, ontologias exercem um papel fundamental no sentido de tornar as informações da internet mais acessíveis utilizando metadados compreensíveis por máquinas. Ontologias compartilhadas

definem um entendimento comum de termos específicos e isto torna possível a comunicação no nível semântico (BOUQUET et al., 2004).

2.2.1 Definição de ontologia

Há várias definições sobre o que é uma ontologia e elas têm mudado ao longo do tempo (CORCHO et al., 2003). Uma das definições mais conhecidas é apresentada por (GRUBER, 1993A), que define ontologia como uma especificação explícita de uma conceituação. Segundo Gruber, uma conceituação é uma visão abstrata e simplificada de um mundo que queremos representar para algum propósito.

Para Borst (1997), uma ontologia é uma especificação formal de uma conceituação compartilhada, sendo formal pelo fato de ser compreensível aos computadores e compartilhado por tratar de conhecimento consensual.

Studer et al. (1998) estabelecem uma nova definição na qual ontologia é uma especificação formal e explícita de uma conceituação compartilhada. Esta definição é uma combinação das definições de Gruber e Borst.

Guarino (1998) discute a definição de Gruber incluindo o aspecto intencional para esclarecer a diferença entre uma ontologia e uma conceituação e estabelece uma visão de construir ontologias usando teoria lógica. Para Guarino uma ontologia é considerada um conjunto de axiomas lógicos que corresponde ao significado intencional de um vocabulário. Dada uma linguagem **L** com compromisso ontológico **K**, uma ontologia para **L** é o conjunto de axiomas descritos de uma maneira que o seu modelo se aproxima o máximo possível do conjunto de modelos intencionais de **L** de acordo com **K**. Em geral, não é uma tarefa fácil encontrar os axiomas corretos.

As diferentes definições são geradas por pontos de vistas diferentes e complementares a uma mesma realidade. Alguns autores criam definições que são independentes do processo, enquanto outras são influenciadas pelo processo de desenvolvimento (CORCHO et al., 2003). A definição de ontologia pode se basear também na complexidade da sua estrutura. Uma ontologia pode descrever uma hierarquia de conceitos ligados por relações de subsunção, conceito mais alinhado às taxonomias; ou uma estrutura, na qual são adicionados os axiomas com o objetivo de expressar relações entre os conceitos e restringir suas interpretações intencionais (GUARINO, 1998).

A comunidade de ontologias faz uma distinção entre ontologias que são principalmente uma taxonomia, daquelas que modelam um domínio de maneira a fornecer restrições sobre a semântica. As ontologias *lightweight* incluem na sua estrutura apenas conceitos, relações e propriedades e as ontologias *heavyweight* adicionam axiomas às ontologias *lightweight* (CORCHO et al., 2003). Esta distinção destaca a diferença de estrutura de conhecimento que cada uma delas pode representar. Enquanto as ontologias mais simples definem uma estrutura de conhecimento, as ontologias mais complexas adicionam o poder de raciocínio à esta estrutura. A capacidade de raciocínio permite a uma ontologia, por meio de expressões lógicas, inferir novos conhecimentos baseados em conhecimentos existentes. Esta característica permite que o comportamento de um sistema computacional não esteja condicionado somente aos fatos representados em um banco de dados.

Diferentemente de estruturas de dados tradicionais, as ontologias podem expressar relações de forma mais restrita entre os conceitos representados. Nesta estrutura complexa, o objetivo principal é a captura de conhecimentos e a reutilização do modelo de um domínio em vez de prover uma estrutura para armazenamento de dados de instâncias (MUSEN, 2002).

2.2.2 Tipos de ontologias

Na literatura, ontologias são classificadas de acordo com diferentes características. Guarino (1998) as caracteriza de acordo com o nível de generalidade do conteúdo representado em ontologias de alto nível, de domínio, de tarefa e de aplicação, conforme Figura 2-1. Estes níveis são descritos a seguir:

- ontologia de alto nível: descreve conceitos ou conhecimentos de senso comum que são independentes de um domínio ou problema particular. As ontologias de alto nível também são conhecidas como ontologia de fundamentação (GUIZZARDI, 2005).
- ontologia de domínio: fornece um vocabulário relacionado a um domínio genérico, especializando termos introduzidos na ontologia de alto nível.
- ontologia de tarefa: descreve uma tarefa ou atividade genérica, especializando também os termos introduzidos na ontologia de alto nível.

- ontologia de aplicação: descreve conceitos dependentes de um domínio ou de uma tarefa particular, sendo que estas ontologias frequentemente estendem ou especializam as ontologias de domínio e de tarefa.

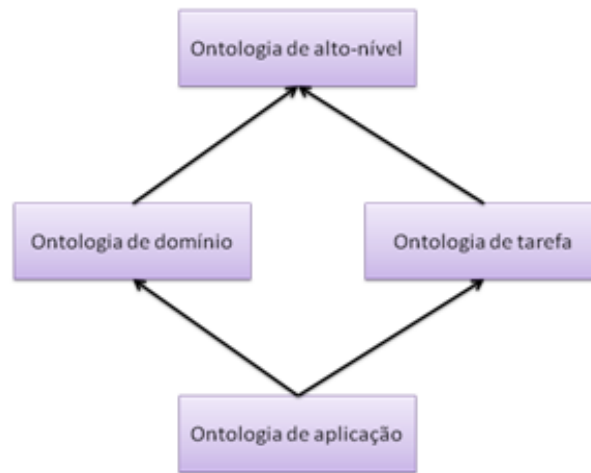


Figura 2-1. Tipos de ontologias de acordo com o conteúdo, adaptado de (GUARINO, 1998)

Em (LASSILA; MCGUINNESS, 2001) ontologias são classificadas de acordo com a “riqueza” de sua estrutura interna. As principais categorias são:

- vocabulários controlados: são as ontologias mais simples, por exemplo, um catálogo.
- glossários: são listas de termos e significados, geralmente expressos em linguagem natural.
- tesouro: provê semântica adicional entre termos e relação de sinônimo, no entanto não provê uma estrutura hierárquica explícita.
- hierarquia informal “*is-a*”: tem uma noção geral de generalização e provê especialização apesar de não ser uma hierarquia de subclasse rigorosa.
- hierarquia formal “*is-a*”: organiza conceitos de acordo com uma hierarquia de subclasses rigorosa.
- frames: incluem classes e suas propriedades e podem ser herdados por classes de nível inferior da taxonomia formal “*is-a*”.
- restrição de valor: permite à aplicação restringir os valores associados a uma propriedade.
- restrições em lógica geral: são geralmente escritas em linguagens lógicas bastante expressivas, permitindo a especificação de restrições em conceitos e propriedades.

Independentemente do tipo da ontologia, a sua aplicação vem se tornando uma tendência em diversas áreas, pois ontologias permitem representações de conhecimentos por meio de estruturas simples ou complexas, com conteúdos mais especializados ou gerais.

2.2.3 Construção de ontologias

Para a construção de ontologias, é necessário definir os métodos ou metodologias, a linguagem para formalizar a ontologia e as ferramentas que apoiam este processo.

Métodos e metodologias podem ser utilizados com o objetivo de melhorar o processo de construção de ontologias. Alguns dos métodos disponíveis são: Uschold e King (USCHOLD; KING, 1995), KACTUS (BERNARAS et al., 1996) e SENSUS (SWARTOUT et al., 1997), enquanto algumas das metodologias disponíveis são: Gruninger e Fox (GRUNINGER; FOX, 1995), 101 (NOY; MCGUINNESS, 2001) e METHONTOLOGY (PEREZ et al., 1996). Cada método e metodologia pode apresentar abordagens distintas no processo de desenvolvimento. Algumas preveem a construção da ontologia desde o seu início, enquanto outras preveem a construção a partir de uma ontologia existente. No entanto, nenhuma abordagem ainda é madura o suficiente quando comparada com as metodologias propostas para a Engenharia de Software e Engenharia do Conhecimento (CORCHO et al., 2003).

A escolha de uma linguagem para a representação de ontologias depende do objetivo que se deseja atingir. Guizzardi (GUIZZARDI, 2007) sugere a necessidade de duas classes de linguagem de representação na engenharia de ontologias. A primeira classe, neste trabalho chamada de “conceitual”, se refere às linguagens filosoficamente bem fundamentadas, focadas na expressividade e na clareza conceitual. A segunda classe, neste trabalho chamada de “computacional”, se refere às linguagens focadas em questões orientadas à computação, tais como, decidibilidade e raciocínio automático.

Considerando a classificação proposta por Guizzardi, pode-se citar a OntoUML (GUIZZARDI, 2005) e a *Unified Modeling Language* (UML) (UML, 2014), como exemplos de linguagens conceituais, pois estão mais focadas na clareza conceitual de um domínio do conhecimento. Enquanto que, linguagens como, o KIF (*Knowledge Interchange Format*) (GENESERETH; FIKES, 1992), OIL (*Ontology*

Inference Layer ou *Ontology Interchange Language*) (HORROCKS et al., 2000) e *Web Ontology Language* (OWL) (DEAN et al., 2002), são consideradas computacionais, pois estão orientadas a questões tais como, processamento automático pelos computadores e raciocínio automático.

A ontologia pode ser inicialmente representada em uma linguagem conceitual, fase em que o desenvolvimento está orientado pelo entendimento de um domínio do conhecimento. Posteriormente, a ontologia pode ser evoluída para uma implementação computacional, quando se espera obter os benefícios do processamento automático pelos computadores.

Para dar apoio às tarefas relacionadas à implementação de uma ontologia, há ferramentas disponíveis conhecidas como editores de ontologias. Para a construção de modelos conceituais bem fundamentados, pode-se citar, por exemplo, as ferramentas: ONTOUML WebEditor (WEBEDITOR, 2014) e a OntoUML *Lightweight Editor* (OLED) (OLED, 2014). Enquanto que para a construção de ontologias orientadas mais para questões computacionais, pode-se citar, por exemplo, as ferramentas: WebOnto (DOMINGUE, 1998), WebODE (ARPÍREZ et al., 2001), OntoEdit (SURE et al., 2002) e Protégé (PROTÉGÉ, 2011).

2.3 Considerações sobre o capítulo

Neste capítulo, foi apresentado de uma maneira abrangente dois conceitos fundamentais para o desenvolvimento desta tese: Engenharia de Requisitos e Ontologias. Ao entender estes dois conceitos é possível perceber que as ontologias podem ser um recurso relevante para resolver problemas clássicos do processo de ERS. No entanto, é importante aprofundar este entendimento e identificar o que tem sido desenvolvido e o que é possível desenvolver neste contexto. O próximo capítulo descreve os principais resultados relacionados ao uso das ontologias na Engenharia de Requisitos.

CAPÍTULO 3 - USO DAS ONTOLOGIAS NA ENGENHARIA DE REQUISITOS: REVISÃO SISTEMÁTICA

Este capítulo apresenta a revisão sistemática realizada com o objetivo de compreender melhor como as ontologias estão sendo usadas no processo da ERS. A revisão também teve como objetivo identificar lacunas de pesquisa e contribuiu significativamente para a motivação desta tese. A revisão iniciou com uma base de 2407 artigos e foi concluída com a análise detalhada de 60 artigos. Os resultados foram publicados em (VALASKI et al., 2016a).

Embora o Capítulo 5 (Estruturação da Pesquisa) tenha o objetivo de detalhar todas as etapas de pesquisa executadas ao longo desta tese, é importante ressaltar que a revisão sistemática foi a 1ª etapa da pesquisa. Esta etapa possibilitou entender melhor o objeto de estudo e identificar as motivações desta tese.

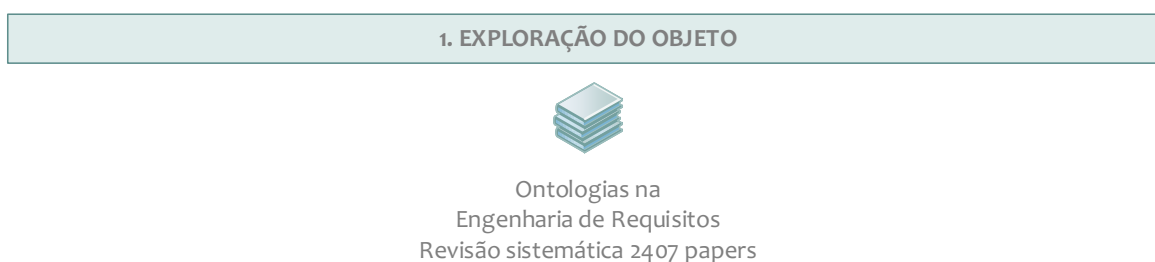


Figura 3-1. 1ª etapa da pesquisa

3.1 Motivação

O uso de ontologias na ERS pode ser motivada por objetivos distintos, como por exemplo: o compartilhamento de um vocabulário comum (ARANDA et al., 2008); o entendimento do domínio do problema (LI, 2005); a análise da expressividade da linguagem (HARZALLAH et al., 2012); a representação do domínio mais próxima do mundo real (ZHANG et al., 2007) e a melhoria da comunicação entre especialistas de diversos domínios (KILOV; SACK, 2009). Considerando a versatilidade das ontologias na solução dos problemas pertinentes a ERS, considerou-se importante identificar como as ontologias estão sendo aplicadas. As questões que nortearam a pesquisa estão apresentadas no Quadro 3-1.

Quadro 3-1. Questões de pesquisa da revisão sistemática.

ID	Questão	Objetivo
QP1	Quais são as atividades da ERS nas quais as ontologias têm sido aplicadas?	Identificar quais têm sido as principais atividades ERS onde as ontologias têm sido aplicadas.
QP2	Quais são as funções das ontologias na ERS?	Identificar quais são as funções das ontologias quando aplicadas na ERS.
QP3	Quais são as linguagens usadas pelas ontologias?	Identificar quais são as classes de linguagem e linguagens usadas para a representação destas ontologias.
QP4	Quais são os domínios do conhecimento representados?	Quais tipos de recursos da ERS são representados pelas ontologias e quais domínios do conhecimento.
QP5	Quais são as contribuições das ontologias para a ERS?	Identificar evidências empíricas relacionadas às contribuições das ontologias para a ERS.

3.2 Método de pesquisa

O método de pesquisa utilizado foi a revisão sistemática de literatura com o objetivo de identificar, avaliar e interpretar as pesquisas disponíveis e relevantes para uma questão em particular (KITCHENHAM et al., 2004). Para a condução desta revisão foram seguidas as seguintes etapas: planejamento da revisão, identificação da pesquisa, seleção de estudos primários e classificação.

3.2.1 Planejamento da revisão

Nesta etapa foi especificado o protocolo com os processos e os métodos para a aplicação da revisão sistemática de literatura. Neste protocolo foi definido o objetivo da revisão, as questões de pesquisa, as principais fontes primárias de estudo e os critérios para inclusão e exclusão dos artigos. O principal objetivo deste estudo foi identificar como as ontologias estão sendo aplicadas na área de ERS. As principais fontes primárias de estudo e critérios para inclusão e exclusão dos artigos são discutidas nas próximas subseções.

3.2.2 Identificação da pesquisa

O objetivo principal de uma revisão sistemática é encontrar o máximo possível de estudos primários relacionados à questão de pesquisa utilizando uma estratégia de busca imparcial. De acordo com esta premissa, foram utilizadas palavras chaves que pudessem identificar o máximo possível de trabalhos relevantes. No Quadro 3-2 é apresentada a *string* utilizada para as buscas. As buscas foram realizadas nas bases científicas digitais: ScienceDirect, SpringLink, IEEE e ACM Digital, nas quais

foram filtradas publicações do tipo *journal* e conferência. A pesquisa foi realizada em janeiro de 2013 e não houve delimitação de período, ou seja, publicações de qualquer ano foram consideradas. Na seleção dos artigos de conferências, foi adicionada à *string* de busca, o filtro "Publication Title": "Requirements Engineering". O filtro teve o objetivo de restringir os artigos a apenas conferências da área de ERS.

Além das conferências indexadas nas bases digitais citadas anteriormente, também foi considerado o Workshop de Engenharia de Requisitos (WER), importante conferência na comunidade de Engenharia de Requisitos. A pesquisa foi realizada no próprio site onde estão publicados todos os artigos da conferência. Os mesmos filtros utilizados nas bases digitais foram utilizados na seleção dos artigos do WER. No total foram retornados 2419 artigos, porém, entre eles, foram identificados 12 duplicados, os quais foram excluídos. Desta maneira, 2407 artigos formaram a base para a seleção dos estudos primários.

Quadro 3-2. String de busca

<p>Ontology AND ("Requirements Development" OR "Requirements Engineering" OR "Requirements Analysis" OR "Requirements Definition" OR "Requirements Modelling" OR "Requirements Elicitation" OR "Requirements Inspection" OR "Requirements Management" OR "Requirements Negotiation" OR "Requirements Process" OR "Requirements Specification" OR "Requirements Traceability" OR "Requirements Validation" OR "Requirements Verification" OR "Requirements Reuse" OR "Software Requirements" OR "Quality Requirements" OR "Non-functional Requirements" OR "Functional Requirements" OR "Late Requirements" OR "Early Requirements")</p>

3.2.3 Seleção de estudos primários

Nesta etapa foi avaliado se os 2.407 estudos primários identificados no estágio anterior forneciam uma evidência direta sobre a questão de pesquisa. Critérios de exclusão foram definidos no protocolo para a seleção adequada dos estudos em três etapas, as quais estão sumarizadas na Tabela 3-1.

Tabela 3-1. Critérios de exclusão

Etapa	Critérios de exclusão	Método	Artigos excluídos
1a	Claramente não é discutido o tema ontologia ou ERS	Leitura de <i>abstract</i> dos 2407 artigos e eventualmente buscas das palavras chaves no artigo. Em caso de dúvida, o artigo foi mantido para avaliação na segunda etapa	2148
2a	Claramente não é discutida a aplicação de ontologia na área de ERS	Com os 259 artigos restantes da etapa anterior, foram realizadas buscas no artigo utilizando a palavra-chave “ontology” com o objetivo de identificar a aplicação de ontologia na ERS. Em caso de dúvida, o artigo foi mantido para avaliação na terceira etapa	123
3a	Não há aplicação do conceito ontologia na área de ERS	Todos os 136 artigos restantes da etapa anterior foram lidos na íntegra para se certificar da aplicação do conceito de ontologia na área de ERS. No caso de artigos, onde a ontologia já havia sido aplicada exatamente nas mesmas circunstâncias, apenas o artigo mais recente foi mantido para avaliação	76

Na primeira etapa foram lidos os resumos de todos os 2407 artigos com o objetivo de eliminar os artigos que claramente não discutiam o tema ontologia ou ERS. Foram eliminados apenas os artigos nos quais não haviam dúvidas de que os temas não estavam sendo tratados. Grande parte dos artigos (2148) foi eliminada nesta primeira etapa.

Na segunda etapa, com os 259 artigos restantes (aproximadamente 11% do total) foi realizada uma pesquisa no artigo pela palavra-chave “ontology” procurando evidências da sua aplicação em algum tema da ERS. Esta estratégia foi utilizada pelo fato do termo ontologia ser mais restritivo do que todos os demais termos utilizados para a ERS. Nesta etapa foram eliminados 123 artigos.

Por fim, na terceira etapa, os 136 artigos restantes (aproximadamente 6% do total) da etapa anterior foram lidos na íntegra de maneira a se certificar da aplicação do conceito de ontologia na área de ERS. Com a execução desta etapa, 76 artigos foram excluídos. Após a execução das três etapas para aplicação dos critérios de exclusão, resultaram 60 artigos (aproximadamente 2,5% do total), os quais foram analisados e sintetizados de acordo com a classificação a seguir.

3.2.4 Classificação

Após a leitura dos artigos selecionados, eles foram classificados de acordo com as categorias, as subcategorias e os critérios apresentados no Quadro 3-3.

Quadro 3-3. Categorias, subcategorias e critérios para a classificação dos resultados.

Categoria	Subcategoria	Critério
Atividade da ERS	Elicitação	Ontologia é aplicada para: apoiar o entendimento do domínio, identificar as necessidades e as restrições do sistema a ser desenvolvido e/ou descobrir requisitos.
	Análise	Ontologia é aplicada para: analisar e/ou documentar os requisitos obtidos na atividade de elicitación, detectar e resolver conflitos entre os requisitos e/ou representar modelos do problema. Uso de linguagens mais formais para a representação dos modelos.
	Especificação	Ontologia é aplicada para: documentar os requisitos obtidos e /ou criar especificação do software a ser desenvolvido. Uso principalmente de linguagem natural.
	Validação	Ontologia é aplicada para: garantir a entrega do que foi solicitado pelo cliente.
	Negociação	Ontologia é aplicada para: apoiar a priorização e a negociação de requisitos e/ou estabelecer acordo sobre os requisitos a serem desenvolvidos.
	Gerenciamento	Ontologia é aplicada para: apoiar o gerenciamento das atividades pertencentes a ERS, como, por exemplo, gerenciamento dos artefatos, rastreabilidade dos requisitos, gerenciamento de mudança, identificação de riscos e etc.
	Não identificada	Não foi possível identificar a atividade.
Função da ontologia		Identificar a função descrita na proposta e categorização.
Classe da linguagem	Conceitual	Ontologia usa linguagem filosoficamente bem fundamentada e foca na expressividade e clareza do conceito representado.
	Computacional	Ontologia usa linguagem focada em questões computacionais tais como raciocínio automático.
Domínio do conhecimento	Domínio da ERS	Ontologia representa conhecimento do domínio da ERS, como, por exemplo, modelos, diagramas, documentos, artefatos e etc.
	Domínio problema	Ontologia representa conhecimento do domínio problema, isto é, domínio para o qual o software será desenvolvido (saúde, leis, financeiro, etc).
	Não identificado	Não foi possível identificar o domínio.
Tipo de avaliação	Empírica	Foi aplicada avaliação empírica, por exemplo, experimento, estudo de caso, survey e etc.
	Não empírica	Foi aplicada avaliação, porém não empírica, por exemplo, exemplos, ilustrações e etc.
	Não aplicada	Avaliação não foi aplicada.
Contexto da avaliação empírica	Acadêmica	Avaliação empírica aplicada em contexto acadêmico, por exemplo, estudantes de uma universidade ou um laboratório de pesquisa.
	Indústria	Avaliação empírica aplicada com funcionários de uma empresa submetidos a situações reais.
Contribuição da ontologia		Contribuições relatadas pelas avaliações empíricas.

3.3 Resultados

A classificação apresentada no Quadro 3-3 foi proposta com o objetivo de prover respostas mais claras às questões de pesquisa estabelecidas no Quadro 3-1. Para uma apresentação mais objetiva, os resultados são apresentados de acordo com as questões de pesquisa.

3.3.1 QP1: Quais são as atividades da ERS nas quais as ontologias têm sido aplicadas?

O objetivo desta questão foi identificar as atividades da ERS onde as ontologias têm sido mais aplicadas. A Tabela 3-2 apresenta os resultados sumarizados relacionados a esta questão.

Tabela 3-2. Atividades da ERS onde as ontologias são aplicadas

Atividade da ERS	ID	Qtde.	%
Análise	[1-31]	31	51,66
Elicitação	[6][32-44]	14	23,33
Especificação	[9][36][41][45-52]	11	18,33
Gerenciamento	[53-57]	5	8,33
Validação	-	0	0,00
Negociação	-	0	0,00
Não identificada	[58-60]	3	5,00

De acordo com os resultados obtidos, a maioria dos estudos encontrados (51,66%) têm proposto o uso de ontologias para a atividade de análise no contexto da ERS. É possível observar uma expressiva aplicação de ontologias para a atividade de modelagem, como, por exemplo: avaliação da qualidade ou expressividade da linguagem de modelagem [3][22-26][29][31]; integração e transformação de modelos [6][10-11][15][19][27-28]; detecção de conflitos ou inconsistências em modelos [8][17-18][20]; validação dos diagramas gerados [5]; representação de padrões de design durante a modelagem de requisitos [16] e modelagem do sistema a ser desenvolvido [2]. Na atividade de análise, as ontologias também têm sido propostas com o objetivo de: prover reuso de atributos [9][12][14]; definir a gramática da linguagem para a representação dos requisitos [1][30]; melhorar o vocabulário e o significado dos elementos do modelo [13][21] e identificar variabilidade nas solicitações dos usuários [4].

A atividade de elicitação foi a segunda com mais propostas encontradas (23,33%). Nesta atividade, as ontologias têm sido propostas com o objetivo de: apoiar o entendimento do domínio [32-35][40-42]; melhorar a comunicação dos stakeholders [33][35][43]; extrair uma lista inicial de requisitos [36][38-39][44]; reusar o conhecimento [37][42]; e identificar objetivos organizacionais [6].

A terceira atividade (18,33%) com mais propostas de ontologias foi a especificação de requisitos. As ontologias têm sido aplicadas com o objetivo de: verificar inconsistências entre os requisitos [36][45-46][49][51]; facilitar integração entre especificação e outras atividades da ERS [51-52]; melhorar como as sentenças são escritas [41][48]; reusar atributos da especificação de requisitos [9][47]; e estruturar os atributos de qualidade [50]. O gerenciamento foi a quarta atividade com mais propostas (8,33%). Nesta atividade as ontologias têm sido aplicadas com o objetivo de: permitir rastreabilidade entre os artefatos produzidos durante o ciclo da ERS [53][55-56]; verificar os riscos associados a segurança de requisitos [54]; e busca pelos artefatos produzidos ao longo do ciclo da ERS [57]. Não foram encontradas propostas aplicadas às atividades de validação e negociação. Também foram encontrados estudos nos quais não foi possível identificar a atividade de aplicação. Nestes estudos as ontologias foram propostas para: representar padrão de requisitos não funcionais [58]; esclarecer o significado do processo de transparência [59]; e estruturar formas de representação de requisitos [60]. A maioria das propostas analisadas foram classificadas em uma única atividade, com exceção das propostas [6][9][36][41].

3.3.2 QP2: Quais são as funções das ontologias na ERS?

O objetivo desta questão foi identificar as principais funções das ontologias quando aplicadas no processo da ERS. Não foi usada uma categoria formal para realizar a classificação. De acordo com as leituras, foram feitas anotações das funções relatadas pelos autores. Ao final das anotações de todos os estudos analisados foi feito um agrupamento das funções afins. Diversas propostas apresentaram mais do que uma função da ontologia. Os resultados são apresentados na Tabela 3-3, de acordo com as sete funções encontradas.

Tabela 3-3. Função das ontologias

Função da ontologia	ID	Qtde.	%
Estruturação e recuperação do conhecimento	[26][29][32][34][38][63-64][66-70][75-78][80]	17	28,33
Verificação e validação	[8][27][30][33][37][39-40][42][56][65-66][68-69][71][74]	15	25,00
Apoio para entender e/ou identificar conceitos	[6][15][32-33][43][46-47][53][60-64][70][79]	15	25,00
Controle de vocabulário	[5][8][32][35][52][54-55][58-59][61-62][65][68][76]	14	23,33
Integração e transformação de modelos	[8][28][31-33][37][41][49-50][66][71-73][75]	14	23,33
Avaliação da linguagem de representação	[5][14][44-48][51-53]	10	16,66
Reúso	[8][26][31][34][36-37][57][62][67][78]	10	16,66

A função mais relatada nos estudos analisados foi a estruturação e recuperação do conhecimento (28,33), enquanto que as funções menos relatadas nos estudos foram: a avaliação da linguagem de representação (16,66%) e o reúso (16,66%). No entanto, é possível observar que não houve muita diferença em termos percentuais entre as categorias encontradas.

3.3.3 QP3: Quais são as linguagens usadas pelas ontologias?

Para responder esta questão, as propostas foram agrupadas de acordo com a classe da linguagem e a linguagem usada para a representação da ontologia. As categorias de classe da linguagem foram utilizadas da proposta de (GUIZZARDI, 2007). Na classe de linguagem conceitual, algumas propostas não relataram o uso de uma linguagem específica. A Tabela 3-4 apresenta os principais resultados relacionados a esta questão.

A classe de linguagem computacional foi a mais usada para a representação das ontologias (55%). A classe de linguagem conceitual foi aplicada em 41,67% das propostas. Duas propostas (3,33%) aplicaram os dois tipos de classe de linguagem. É importante ressaltar que os resultados mostraram uso expressivo de ambas as classes de linguagem para a representação de ontologias no processo de ERS.

Nas propostas que usaram a classe de linguagem computacional, a linguagem OWL foi a mais aplicada. O mecanismo de inferência desta linguagem é uma das justificativas para o seu uso. Estes mecanismos são aplicados

principalmente para prover interoperabilidade, reúso e identificação automática de inconsistências. Alguns exemplos de objetivo das ontologias que usaram a classe de linguagem computacional foram: permitir interoperabilidade e validação dos modelos gerados[8]; eliminar ambiguidades em documentos [61]; prover reúso de atributos de qualidade em especificação de requisitos [31]; ajudar os stakeholders na construção de uma especificação mais completa, consistente, sem ambiguidade e rastreável [71]; extrair conhecimento para ser utilizado em modelagem de negócios [36]; permitir a transformação de modelos em modelos lógicos [41]; propiciar rastreabilidade entre os artefatos produzidos [75]; e promover uma estrutura comum que facilite a localização dos artefatos produzidos durante o processo da ERS [77].

Tabela 3-4. Classe e linguagens usadas pelas ontologias

Classe	Linguagem/Modelo	ID	Qtde	%
<i>Computacional</i>			33	55,00
	OWL	[8][29][31][36][38-41][60-61][71][75][77]	13	
	RDF/OWL	[27][34]	2	
	RDF	[32][59]	2	
	Alchoin	[42]	1	
	FCA (Formal Concept Analysis)	[74]	1	
	FOL (First-order Logic)	[76]	1	
	Predicate logic	[55]	1	
	Temporal logic	[66]	1	
	Prolog	[6]	1	
	SIN (Description Logic)	[72]	1	
	XML	[68]	1	
	Não informado	[30][37][56-57][65][67][70][73]	8	
<i>Conceitual</i>			25	41,67
	Bunge Model	[43][47-49][51][53]	6	
	BWW Ontology	[14][44-45]	3	
	UML	[28][62]	2	
	AORML	[46]	1	
	DOLCE	[52]	1	
	i* Framework	[64]	1	
	StarUML	[78]	1	
	Não informado	[5][15][26][35][50][54][58][63][79-80]	10	
<i>Híbrida</i>			2	3,33
	UML/OWL	[33]		
	i*; KAOS/Frame	[69]		

Já nas propostas que usaram a classe de linguagem conceitual, o modelo de Bunge e a ontologia de Bunge-Wand Weber (BWW) , que também é baseado no modelo de Bunge, foram os mais usados. Nestas propostas, as ontologias foram aplicadas principalmente com o objetivo de avaliar a qualidade e a expressividade das linguagens e dos modelos usados para a representação de ontologias. A ontologia BWW tem sido usada, por exemplo, com o objetivo de: avaliar a qualidade da linguagem *Artificial Intelligence Markup Language* (AIML) [14]; identificar deficiências da linguagem [44]; e avaliar as técnicas de modelagem e como compará-las [45]. O modelo de Bunge tem sido aplicado com o objetivo de: sugerir significados mais precisos para os elementos do modelo conceitual e definir semântica formal para a linguagem de modelagem conceitual [43]; avaliar a sintaxe da linguagem de modelagem [47]; identificar deficiências na gramática usada pelos diagramas de entidade/relacionamento [48]; e avaliar os modelos de processo existentes [51]. Algumas propostas não informaram a linguagem utilizada.

3.3.4 QP4: Quais são os domínios do conhecimento representados?

Esta questão teve o objetivo de classificar os domínios do conhecimento representados pelas ontologias. Ao longo da revisão foi possível observar duas categorias principais de domínios de conhecimento: domínio da ERS e domínio do Problema. O domínio da ERS envolve a representação de artefatos, métodos, modelos e técnicas da área de Engenharia de Requisitos para apoiar as atividades desenvolvidas ao longo do processo de ERS. Já o domínio do Problema envolve a representação do conhecimento da área para a qual o software será desenvolvido (saúde, educação, legislação, etc). Sendo assim, os estudos foram agrupados nestas duas categorias de domínio do conhecimento. A Tabela 3-5 apresenta os principais resultados relacionados a esta questão.

Tabela 3-5. Domínio do conhecimento representado pelas ontologias

Domínio do conhecimento	ID	Qtde	%
ERS	[1-3][8-9][12-31][44][46-60]	41	68,33
Problema	[4-7][10-11][32-40][42-43][45]	18	30,00
Outro	[41]	1	1,67

A maioria das propostas analisadas (68,33%) usaram ontologias para a representação de recursos pertinentes ao processo da ERS. Para entender melhor quais são estes recursos, a Tabela 3-6 apresenta um agrupamento destes recursos.

Tabela 3-6. Domínio do conhecimento da ERS representado pelas ontologias

Recurso da ERS	Subtipo	ID	Qtde	%
Modelo			21	51,22
	Modelo de negócios	[14][20][22][27][29]		
	Geral	[15][19][25][28]		
	Modelo conceitual	[3][21][24]		
	Diagrama de atividades	[8][23]		
	Caso de uso	[18][48]		
	Diagrama de classe	[31]		
	Colaborativo	[2]		
	Diagrama entidade/relacionamento	[26]		
	Padrões	[16]		
	Família de produtos	[12]		
RNF			11	26,83
	Qualidade	[9][17][50]		
	Acessibilidade	[47]		
	Confidencialidade	[49]		
	Geral	[46]		
	Padrões	[58]		
	Segurança	[54]		
	Transparência	[59]		
	Confiança	[13]		
	Usabilidade	[44]		
Artefatos geral			4	9,75
	-	[53][55-57]		
Conceitos <i>core</i> ERS	-		4	9,75
		[1][30][52][60]		
Documento			1	2,44
	Especificação	[51]		

De uma maneira geral, os modelos (51,22%) são os recursos do domínio da ERS com mais propostas de aplicação das ontologias. Uma das possíveis justificativas para a sua significativa aplicação é que modelos e ontologias têm o objetivo em comum de representação formal de um conhecimento. Alguns exemplos de propostas encontradas nesta categoria são: modelo para processos colaborativos [14]; ontologia para metodologia de desenvolvimento de sistema orientado a agentes [15]; ontologia para formalizar palavras-chave de um modelo conceitual [19], ontologia para representar fluxos de exceção em modelos de processo de negócio [20]; uso de conceitos ontológicos para a AORML (*Agent-Object Relationship Modeling Language*) [24]; e ontologia para identificar componentes de um modelo de processo [27].

Também foi encontrado um número expressivo de propostas (26,83%) para a representação de requisitos não funcionais, principalmente para requisitos não

funcionais de qualidade. Também foram encontradas ontologias para representação de artefatos (9,75%) com o objetivo de rastreabilidade e busca, ontologias para a representação de conceitos *core* que envolvem a área da SRE (9,75%) e apenas uma ontologia para a representação de documento de especificação de requisitos (2,44%).

3.3.5 QP5: Quais são as contribuições das ontologias para a ERS?

Esta questão teve o objetivo de identificar evidências relacionadas à contribuição do uso de ontologias no processo da ERS. Para responder esta questão, foram identificados o tipo de avaliação apresentado no estudo, o contexto da aplicação da avaliação e por fim, para as avaliações empíricas, foram extraídas as contribuições.

Primeiramente as propostas foram classificadas de acordo com o tipo da avaliação. A Tabela 3-7 apresenta o resultado da classificação.

Tabela 3-7. Tipo de avaliação aplicada nos estudos

Tipo de avaliação	ID	Qtde	%
Não empírica	[1-2][4][6-8][10][12-14][16][21-23][25][28-30][34-35][37][39][42][44][46-48][51-52][54][57-59]	33	55,00
Empírica	[3][9][11][15][17-20][26-27][31-33][36][38][40-41][43][45][49-50][55-56]	23	38,33
Sem avaliação	[5][24][53][60]	4	6,67

Em termos gerais três grupos foram identificados: avaliação não empírica (55,00%), avaliação empírica (38,33%) e sem avaliação (6,67%). Este resultado mostra que área da ERS demanda por mais trabalhos empíricos, a partir dos quais seja possível tirar conclusões mais sólidas.

Entre as avaliações empíricas, foi identificado o contexto da avaliação, classificado em: acadêmico e indústria. Embora ambos os contextos sejam importantes para a extração de evidências, o contexto de indústria frequentemente apresenta condições mais próximas do problema a ser analisado. A Tabela 3-8 mostra que a maioria das avaliações empíricas (78,26%) descritas nos estudos ocorreram em contexto acadêmico, enquanto apenas (21,74%) das avaliações empíricas ocorreram em contexto de indústria. Ontologia ainda é um formalismo mais disseminado na academia do que na indústria. Resultados mais significativos

poderão ser obtidos quando houver maior número de avaliações empíricas em contexto de indústria.

Tabela 3-8. Contexto da avaliação empírica aplicada nos estudos

Contexto	ID	Qtde	%
Academia	[3][11][17-20][26][31-32][36][38][40-41][43][49-50][55-56]	18	78,26
Indústria	[9][15][27][33][45]	5	21,74

Além do contexto, para cada estudo que apresentou avaliação empírica, foi extraída a contribuição da ontologia para o problema exposto. Todas as contribuições foram anotadas e posteriormente agrupadas em categorias. A Tabela 3-9 apresenta estes resultados.

Tabela 3-9. Contribuição da ontologia aplicada nos estudos

Contribuição da ontologia	ID
Identificar problemas em especificação e modelos	[11][15][18-20][45][49]
Melhorar a comunicação	[3][32-33][43]
Construir modelos mais completos	[3][17][40-41]
Permitir a rastreabilidade entre os artefatos	[55-56]
Identificar requisitos de qualidade	[36][50]
Alinhar negócios e sistema	[27]
Melhorar a compreensão do domínio	[31]
Melhorar a expressividade dos modelos	[26]
Apoiar a transformação de modelos	[9]
Sem contribuição positiva	[38]

A categoria de contribuição “identificar problemas em especificação de modelos” foi uma das mais citadas entre os estudos que apresentaram avaliação empírica. Neste agrupamento, as contribuições foram relatadas da seguinte maneira: identificar inconsistências nos requisitos [11]; identificar entidades nas especificações [11]; identificar detalhes omitidos na especificação [15][18]; verificar inconsistências nos modelos [19]; identificar violações na especificação [20]; detectar informação faltante em cenários [45]; e identificar especificações incompletas e ambíguas [49].

As categorias “melhorar a comunicação” e “construir modelos mais completos” foram a segunda categoria mais citadas. A categoria “melhorar a comunicação” foi reportada como: melhorar a comunicação entre os stakeholders [3][33][43] e melhorar o entendimento entre os desenvolvedores do software [32]. A

categoria “construir modelos mais completos” foi relatada como: identificar o construto correto da linguagem [3]; melhorar a qualidade dos modelos construídos por novatos [17]; construir modelos mais precisos [40]; e construir modelos mais completos e precisos [41].

A categoria “permitir rastreabilidade entre os artefatos” foi relatada em ambas as propostas [55-56] como: apoiar a rastreabilidade de artefatos produzidos durante o processo da ERS. A categoria “identificar requisitos de qualidade” foi relatada como: melhorar a identificação dos requisitos de qualidade [36] e apoiar o analista a identificar atributos de qualidade [50]. Para as demais categorias foi identificada apenas uma proposta. É importante destacar que houve uma proposta que relatou resultado sem contribuição positiva. Em [38] o experimento mostrou que a ontologia não apresentou resultados positivos para apoiar a aquisição de requisitos no domínio jurídico usando documentos regulatórios.

3.4 Considerações sobre a revisão sistemática

As ontologias têm apresentado contribuições consideráveis com relação a solução de problemas relacionados ao processo da ERS. A revisão sistemática foi realizada com o objetivo de obter uma visão expandida das propostas mais recentes e identificar lacunas para contribuição científica.

Os resultados mostraram trabalhos efetivos principalmente para apoiar as atividades de análise, especificação e elicitação. Não foram encontradas propostas relacionadas às atividades de negociação e validação. As contribuições mais significativas das ontologias no processo da ERS têm sido: identificar problemas nas especificações e modelos, melhorar a comunicação e construir modelos mais completos. Também foi observada maior aplicação empírica dos estudos em contexto acadêmicos, o qual sugere que os benefícios do uso das ontologias ainda não estejam disseminados na indústria.

Entre as classes de linguagem usadas, a classe conceitual é mais empregada quando o objetivo ainda não é a implementação do software, mas o entendimento do domínio do problema para o qual o software será desenvolvido. Por outro lado, a classe computacional tem maior aplicação quando os artefatos são produzidos durante o processo da ERS. Há uma expressiva concentração de propostas aplicando ontologias à modelos (conceitual, negócios, atividades, processo, etc.).

Modelos têm sido importantes recursos não somente na atividade de análise, mas também na transformação de modelos para a fase de design do software.

O resultado pertinente ao uso de linguagem com primitivas ontológicas para a representação de modelos conceituais, foi a base principal para o desenvolvimento desta tese de doutorado. As ontologias quando aplicadas usando linguagem conceitual, têm como principal objetivo prover o entendimento de um domínio por meio de modelos conceituais considerando questões relativas a expressividade da linguagem aplicada. As propostas mostram a importância em se utilizar técnicas e linguagens de modelagem ontologicamente fundamentadas. Os conceitos de um universo de discurso são entidades abstratas que existem apenas na mente dos usuários. Para que eles sejam capturados, eles precisam ser representados em um artefato concreto, ou seja, uma especificação. Isto implica, o uso de uma linguagem para representá-los de maneira concisa, completa e não ambígua. Uma especificação produzida deve ser uma representação compartilhada de entidades de um domínio que os especialistas consideram relevantes em um universo de discurso, o qual pode ser utilizado para promover a solução de problemas, comunicação, aprendizagem e reuso em alto nível de abstração (ARANGO, 1994). O uso de uma linguagem com falhas de expressividade pode comprometer o entendimento dos artefatos de requisitos em fases posteriores.

Mylopoulos (MYLOPOULOS, 1992) define a modelagem conceitual como uma atividade de descrever formalmente aspectos do mundo físico e social com o propósito de entendimento. A adequação de uma notação de modelagem conceitual está baseada na sua contribuição para a construção de modelos da realidade que promovam um entendimento comum da realidade entre seus usuários humanos (MYLOPOULOS, 1992). Considerando esta perspectiva, torna-se cada vez mais necessário o uso de linguagens com primitivas ontologicamente bem fundamentadas que ajudem a representar o máximo possível a realidade do domínio de um problema (GUIZZARDI, 2005). Uma vez que os modelos conceituais ontologicamente bem fundamentados proveem uma representação mais próxima da realidade de um domínio, alguns dos problemas recorrentes da atividade de requisitos podem ser minimizados com a intensificação do seu uso. Segundo Falbo et al. (FALBO et al., 1998) um modelo de domínio pode ser usado na aplicação da análise de requisitos para melhorar a comunicação e o entendimento do domínio e ajudar no processo de elicitación de requisitos. Desta maneira, no escopo desta

pesquisa, considera-se que modelos conceituais ontologicamente bem fundamentados podem ser um instrumento importante a ser utilizado na atividade de elicitação de requisitos.

3.5 Considerações sobre o capítulo

Neste capítulo, foram apresentados os principais resultados obtidos por meio de revisão sistemática de literatura relacionados à aplicação das ontologias como solução para problemas do processo da ERS. Os resultados desta revisão definiram a base da proposta da tese. Como a expressividade da linguagem influencia diretamente na comunicação do modelo, o uso de uma linguagem adequada se torna essencial para representar o domínio do problema para o qual o software será desenvolvido. No próximo capítulo, serão apresentados os principais conceitos relacionados à linguagem OntoUML, linguagem definida, no escopo desta pesquisa, como instrumento essencial para melhorar o entendimento do domínio e conseqüentemente a elicitação de requisitos de software.

A Figura 3-2 resume a etapa da pesquisa em questão e o principal resultado.

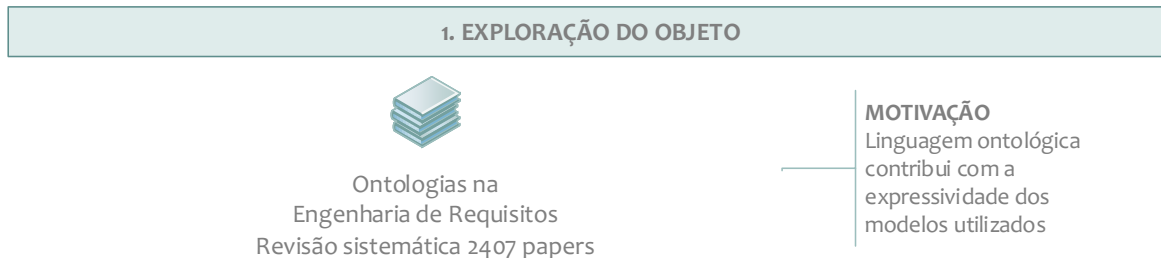


Figura 3-2. 1ª etapa da pesquisa e resultado

CAPÍTULO 4 - ONTOUML

De acordo com a revisão sistemática apresentada no capítulo anterior, o uso de linguagem ontologicamente fundamentada tem um papel fundamental na expressividade de um modelo conceitual. Como a base principal desta tese é o uso de modelos conceituais para apoiar o processo de elicitação de requisitos, a escolha da linguagem adequada é essencial. A OntoUML atualmente é uma das linguagens com maior embasamento ontológico quando comparada com outras linguagens, até bem mais populares, como por exemplo, a UML. Esse capítulo apresenta os principais conceitos e construtos relacionados à OntoUML.

4.1 Objetivo

A linguagem OntoUML, proposta por Guizzardi (GUIZZARDI, 2005), foi motivada pela necessidade de uma linguagem fundamentada ontologicamente que provesse a semântica necessária para a construção de modelos conceituais que representasse conceitos fiéis à realidade. Segundo Guizzardi, um dos principais fatores de sucesso por trás do uso de uma linguagem de modelagem é a sua habilidade em prover aos usuários um conjunto de primitivas de modelagem que possam expressar diretamente importantes abstrações de um domínio. O principal objetivo da OntoUML é atingir o comprometimento ontológico falho em outras linguagens genéricas para modelagem conceitual. Para alcançar este objetivo, a OntoUML foi baseada em teorias da metafísica, ciências cognitivas e linguística.

Guizzardi discute em sua proposta os problemas clássicos de modelagem conceitual provenientes da falta de adequação da linguagem utilizada, como por exemplo, a representação de papéis com múltiplos tipos e a transitividade em relações todo-parte. Em modelagem conceitual, os conceitos de todo-parte são geralmente compreendidos superficialmente e, conseqüentemente, estes conceitos são representados com uma axiomatização mínima do que o conceito requer. Guizzardi também discute os problemas de interoperabilidade semântica na integração de ontologias *lightweight*. Além de resolver problemas clássicos da modelagem conceitual, a proposta de Guizzardi foca em dois aspectos principais: a

facilidade do usuário da linguagem reconhecer o que os construtos da linguagem querem dizer, e a facilidade de compreender, comunicar e raciocinar com a especificação produzida nesta linguagem.

As classes propostas na OntoUML são especializações das classes abstratas da *Unified Foundational Ontology* (UFO) e estendem o metamodelo original da UML. Uma ontologia de fundamentação tem o objetivo de prover semântica para linguagens genéricas de modelagem conceitual e assim restringir as possíveis interpretações de suas primitivas de modelagem (GUIZZARDI, 2005). Apesar da OntoUML representar diversos tipos de categorias, considerando o escopo desta pesquisa, serão apresentados com maior destaque apenas os principais construtos pertinentes a categoria objeto.

Na categoria objeto são propostos os construtos relacionados à modelagem conceitual estática de um domínio (GUIZZARDI et al., 2011). A seguir são apresentados os construtos da OntoUML a partir da classe *Universal*. Na filosofia há duas categorias gerais aplicadas a modelagem conceitual, *Universals*, construto relacionado a *Types* (Classes) e *Individuals*, relacionado a suas instâncias. Na Figura 4-1 são apresentadas as categorias *Universal* que divide-se em: *Substantial Universal*, *Moment Universal* e *Relation*.

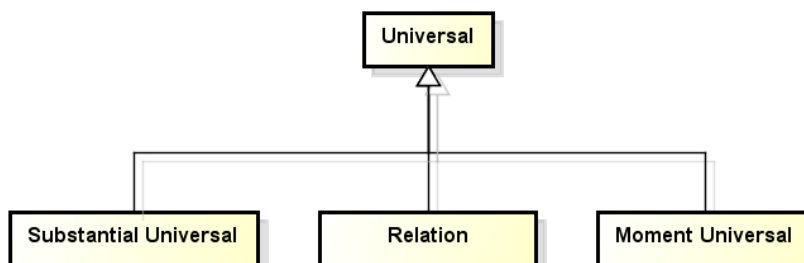


Figura 4-1. Construto Universal e suas subclasses

4.2 Construtos: substantial universal

Substantial, também conhecido em algumas linguagens como *Thing*, *Endurant* ou *Continuant*, são popularmente conhecidos como objetos. *Substantial* são entidades que persistem no tempo enquanto mantêm a sua identidade, oposto aos eventos, como, por exemplo, um processo de negócio ou uma festa de aniversário. *Substantial* inclui entidades persistentes físicas e sociais, como, por exemplo, estudante, bola, pedra e rainha Elizabeth. *Substantial Universal* devem ser

interpretados como *Object Universal*. A classe *Substantial Universal* divide-se em *SortalUniversal* e *MixinUniversal*. A Figura 4-2 apresenta a tipologia da classe *Substantial Universal* e suas subclasses. As classes em cinza representam os estereótipos propostos na OntoUML.

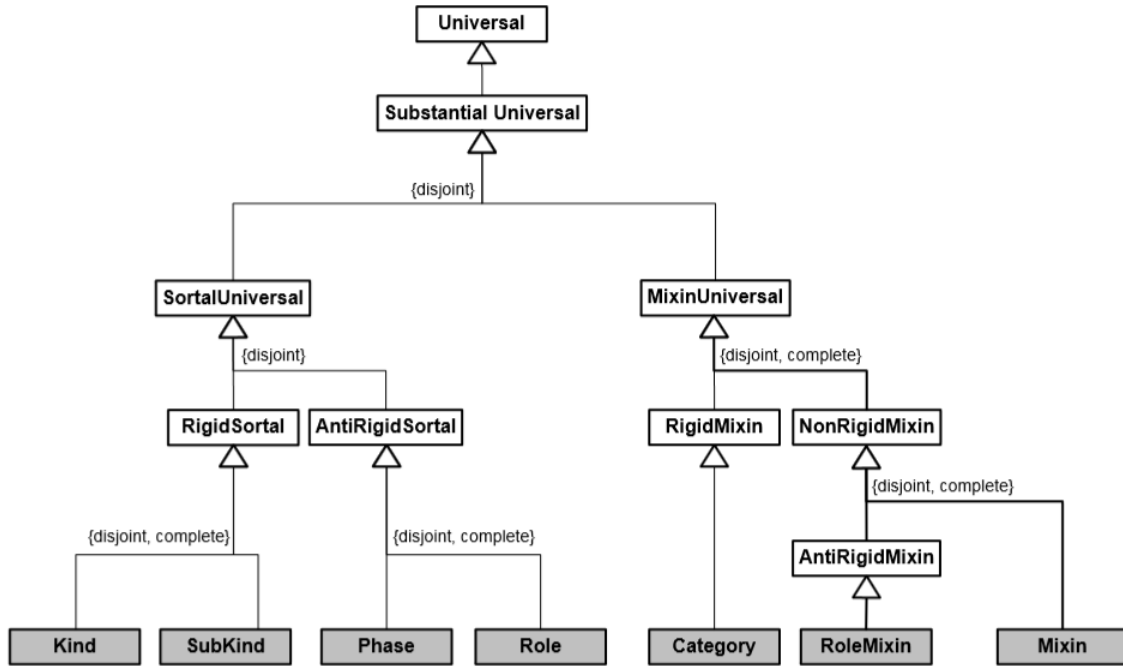


Figura 4-2. Tipologia da classe *Substantial Universal* (GUIZZARDI, 2005)

4.2.1 Sortal Universal

As entidades definidas como *sortal* carregam o princípio da identidade e individualização para as suas instâncias. Cada indivíduo em um modelo conceitual deve ser uma instância de uma entidade representada por um *Sortal Universal*. Um indivíduo não pode ser representado por mais que um *Sortal Universal* distinto.

A classe *Sortal Universal* especializa-se nos tipos *Rigid Sortal* e *AntiRigid Sortal*. Um *sortal* é dito rígido se ele necessariamente se aplica a todas as suas instâncias em todos os mundos possíveis. Enquanto que um *sortal* é dito antirígido se ele necessariamente não se aplica a todas as suas instâncias.

Entre o tipo *Rigid Sortal* estão as categorias *Kind* e *Subkind*. Um *Kind* é um *Rigid Sortal*, e, portanto possui, propriedades intrínsecas, materiais, que provêm claros princípios de identidade e de individualização, rígidas, e que determinam classes de coisas ou seres existencialmente independentes e que são ditos complexos funcionais. Podem ser tanto naturais (cachorro, pessoa) quanto artefatos

(televisão, casa). Um *Subkind* também é um tipo rígido que traz o princípio de identidade e que possui algumas restrições estabelecidas entre o construto *Kind*: a) todo objeto em um modelo conceitual deve ser uma instância de apenas um *Kind*; e b) *Subkind* de um *Kind* deve aparecer na forma de uma partição disjunta.

Para exemplificar os conceitos referentes a classe *Sortal Universal*, é utilizada a ontologia que representa o domínio da Genealogia, apresentada na Figura 4-3. Na ontologia a classe “*Person*” é representada como sendo do tipo *Kind* pois apresenta princípios claros de identidade e individualização. Em qualquer mundo a ser representado, uma instância de uma pessoa sempre será uma pessoa. Já as classes “*Man*” e “*Woman*” são representadas como sendo do tipo *Subkind*, pois também apresentam as propriedades de um *Rigid Sortal* e apresentam uma partição disjunta da classe “*Person*”.

Com relação ao tipo *AntiRigid Sortal*, há duas subcategorias: *Phase* e *Role*. Em ambas, as instâncias podem mudar seus tipos sem afetar a sua identidade. Enquanto que no construto *Phase*, as mudanças podem ocorrer devido às mudanças em suas propriedades intrínsecas, no *Role* as mudanças ocorrem devido às propriedades relacionais. Utilizando a ontologia da Genealogia, tem-se as classes “*LivingPerson*” e “*DeceasedPerson*”, representadas como sendo do tipo *Phase*, e as classes “*Parent*”, “*Father*”, “*Mother*”, “*Offspring*”, “*Ancestor*”, “*Descendent*” representadas como sendo do tipo *Role*, todas relacionadas à classe “*Person*”. A alteração da propriedade intrínseca “estar vivo” é o que causa a mudança de suas fases. Enquanto que, por exemplo, a classe “*Ancestor*” depende da relação com a classe “*Descendent*” e a classe “*Parent*” depende da relação com a classe “*Offspring*”.

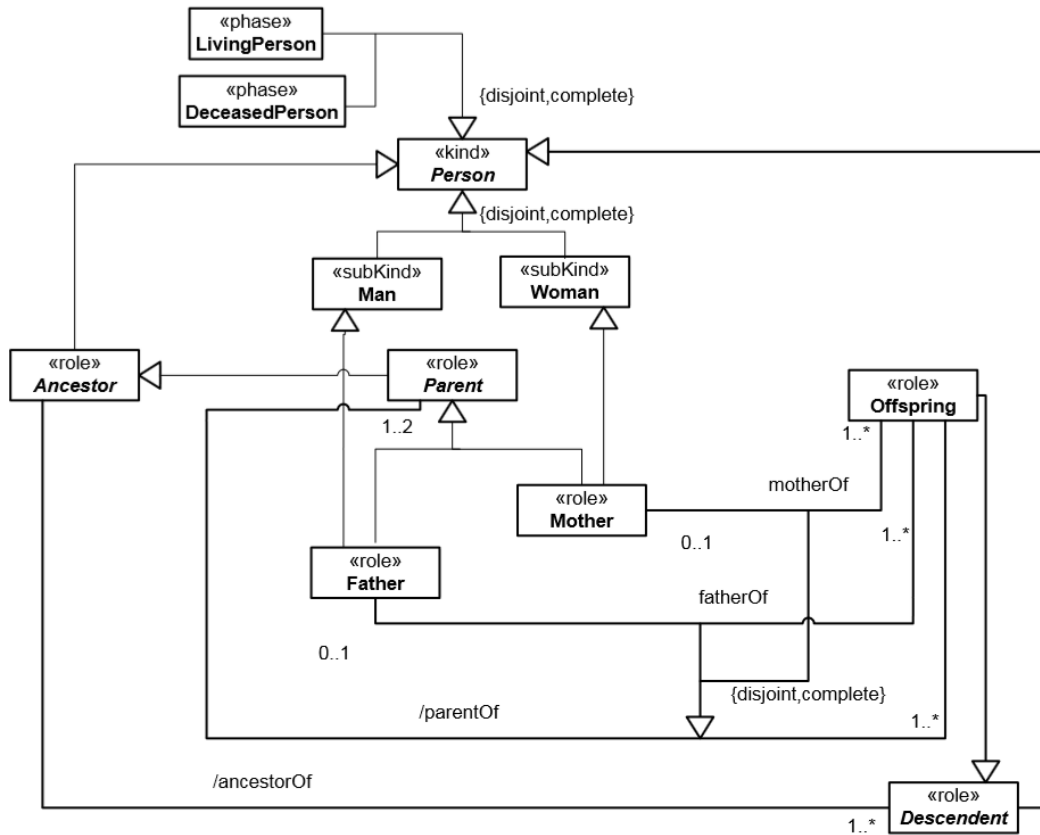


Figura 4-3. Ontologia para o domínio da Genealogia (GUIZZARDI, 2005)

Na proposta de Guizzardi, a classe *Kind* é dividida em três categorias ontológicas distintas: *quantity*, *collective* e *functional complex*, conforme Figura 4-4. Como o tipo *functional complex* é o mais comum entre os *Kinds*, por *default* as classes identificadas por *Kind*, representam o tipo *functional complex*.

A classe *Quantity*, assim como o *Kind*, é um *Substance Sortal* que possui propriedades intrínsecas e materiais, que determinam princípios de identidade de coisas ou seres independentes, cujas instâncias são entidades que correspondem ao conceito de massa tais como água, petróleo e vinho.

A classe *Collective*, assim como o *Kind* e o *Quantity*, também é um *Substance Sortal*. Suas instâncias são coleções de complexos que formam uma estrutura uniforme, como, por exemplo, uma pilha de tijolos. A relação entre um *Collective* e os complexos chama-se “constituição”. No exemplo citado anteriormente, um muro é constituído de tijolos. A classe *Collective* possui o atributo *extensional*, o que significa que todas as suas partes são consideradas essenciais. Assim, a inclusão

ou a remoção de uma dessas partes destrói o *Collective*. As classes em cinza representam os estereótipos da OntoUML.

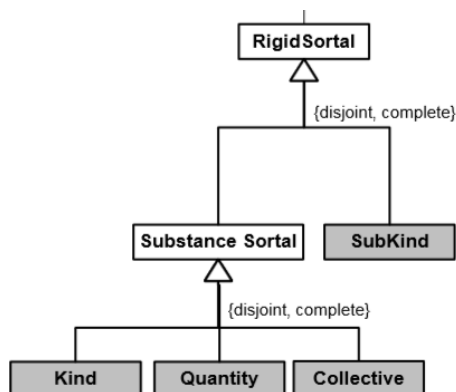


Figura 4-4. Categorias da classe Substance Sortal

4.2.2 Mixin Universal

Mixin Universal é a segunda subclasse de *Substantial Universal* e representa os tipos dispersivos. Os dispersivos são entidades que cobrem conceitos com princípios de identidade diferentes e, portanto, não são sortais. Todos os elementos que são especializados de um sortal, carregam o princípio da identidade. O *Mixin Universal* representa propriedades abstratas que são comuns a tipos diferentes. A classe “*MixinUniversal*” especializa-se em três subclasses: “*Category*”, “*RoleMixin*” e “*Mixin*”.

A classe “*Category*” é uma subclasse de “*RigidMixin*” que subsume diferentes *Kinds*. Pode-se citar como um exemplo desta situação, a classe “*GeographicalSpace*”, criada para representar as propriedades de latitude, longitude e altitude. A partir da classe “*GeographicalSpace*” são especializadas as classes “*SurgeryRoom*”, “*Gallery*” e “*Museum*”, todas do tipo *Kind* porém, cada qual, com princípios de identidade distintos.

Alguns *mixins* são antirígidos pois representam propriedades comuns e abstratas de entidades do tipo *Role*. Estes *mixins* são representados por meio da classe “*RoleMixin*”, subclasse de “*AntiRigidMixin*”. Pode-se citar como exemplo a classe “*Customer*”, apresentada na Figura 4-5. A partir da classe “*Customer*” há duas especializações representadas pelas subclasses “*PersonalCustomer*” e “*CorporateCustomer*”, todas do tipo *Role*, porém, cada qual com princípios de

identidade distintos uma vez que a classe “*PersonalCustomer*” é uma especialização de “*Person*”, e a classe “*CorporateCustomer*” é uma especialização da classe “*Organization*”.

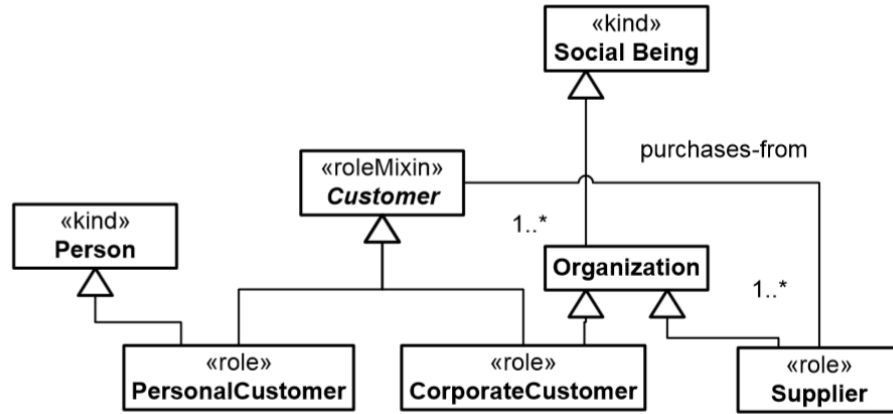


Figura 4-5. Aplicação de um RoleMixin (GUIZZARDI, 2005)

Além disso, há alguns *mixins* que representam propriedades que são essenciais para algumas instâncias e acidentais para outras. Um exemplo é o *mixin* “Sentável” que pode ser considerada uma propriedade essencial para a cadeira, mas acidental para uma caixa. Estes *mixins* são representados pela classe “*Mixin*” subclasse de “*NonRigidMixin*”. O Quadro 4-1 resume os principais estereótipos utilizados na OntoUML referente à tipologia *Substantial Universal*.

Quadro 4-1. Estereótipos da OntoUML referentes aos Substantial Universals

Estereótipo	Descrição
<<kind>>	Um <<kind>> representa um <i>Substantial Sortal</i> cujas instâncias são complexos funcionais. Exemplos incluem tipos naturais (Pessoa, Cachorro, Arvore) e artefatos (Cadeira, Carro, Televisão).
<<quantity>>	Um <<quantity>> representa um <i>Substantial Sortal</i> cujas instâncias são quantidades. Exemplos: coisas que tipicamente, em linguagem natural, se referem a massas (Ouro, Agua, Areia, Refrigerante).
<<collective>>	Um <<collective>> representa um <i>Substantial Sortal</i> cujas instâncias são coletivas, ou seja, são coleções complexas que têm uma estrutura uniforme. Ex: floresta, grupo de pessoas. O tipo <i>collective</i> geralmente se relaciona a um complexo via uma relação de constituição. Ex: um grupo de pessoas constitui um time de futebol. Neste caso, há uma extensão do princípio de identidade.

<p style="text-align: center;"><<subkind>></p>	<p>Um <<subkind>> é um tipo rígido, relacionalmente independente, pois carrega o princípio de identidade herdado de um <i>Substance Sortal</i>. Ex: Homem e Mulher. O estereótipo <<subkind>> pode ser omitido em um modelo conceitual sem perder a clareza.</p> <p>Um <<subkind>> não pode ter como super classe membros das classes <<phase>>, <<role>> e <<roleMixin>>.</p>
<p style="text-align: center;"><<phase>></p>	<p>Um <<phase>> é um tipo antirígido, relacionalmente independente e definido como parte de uma partição e um <i>Substance Sortal</i>. Ex: Lagarta e Borboleta que são partições do <i>kind</i> Lepidóptero.</p> <p><i>Phases</i> são antirígidos, portanto, em um modelo conceitual não podem aparecer como super classe de tipos rígidos (<i>kind</i>, <i>subkind</i>, <i>collective</i> e <i>quantity</i>). A partição de um <i>Substance Sortal</i> é representada no modelo, como sendo um conjunto disjunto e completo de uma generalização. Ou seja, as propriedades <i>isCovering</i> = true e <i>isDisjoint</i> = true são usadas para representação ontológica do conceito.</p>
<p style="text-align: center;"><<role>></p>	<p>Um <<role>> representa um antirígido, relacionalmente dependente de um Universal. Ex: um Estudante é uma instância de um <i>kind</i> Pessoa e deverá estar relacionado a uma Instituição de Ensino.</p> <p><i>Roles</i> são antirígidos, portanto, em um modelo conceitual não podem aparecer como super classe de tipos rígidos.</p>
<p style="text-align: center;"><<category>></p>	<p>Um <<category>> representa um mixin rígido e relacionalmente independente, ou seja, um universal dispersivo que agrega propriedades comuns e essenciais a diferentes <i>Substance Sortal</i>. Ex: uma categoria Entidade Racional como generalização de Pessoa e <i>Agente Inteligente</i>.</p> <p><i>Uma classe do tipo <<category>> só pode ser subclasse de outro <<category>> ou de um <<mixin>>.</i></p>
<p style="text-align: center;"><<roleMixin>></p>	<p>Um <<roleMixin>> representa um mixin antirígido e externamente dependente de um não sortal, ou seja, um universal dispersivo que agrega propriedades que são comuns a diferentes roles.</p>
<p style="text-align: center;"><<mixin>></p>	<p>Um <<mixin>> representa propriedades que são essenciais para alguns e acidentais para outros. Ex: a propriedade "Sentável", a qual representa uma propriedade que pode ser essencial para o <i>kind</i> Cadeira, mas acidental para uma Caixa.</p> <p>Uma classe do tipo <<mixin>> não pode ser subclasse de um <<roleMixin>>.</p>

4.3 Construtos: moment universal e relation

Enquanto os *Substantial Universal* possui um alto grau de independência, os *Moments* são existencialmente dependentes de outros indivíduos. Uma importante característica que identifica todos os *moments* é que eles somente podem existir em outros indivíduos. A classe *Moment Universal* divide-se em *IntrinsicMomentUniversal* e *RelatorUniversal*. A Figura 4-6 apresenta a tipologia da classe *Moment Universal* e suas subclasses, bem como os tipos de relações que podem ser representadas a partir de *Substantial* e *Moment*.

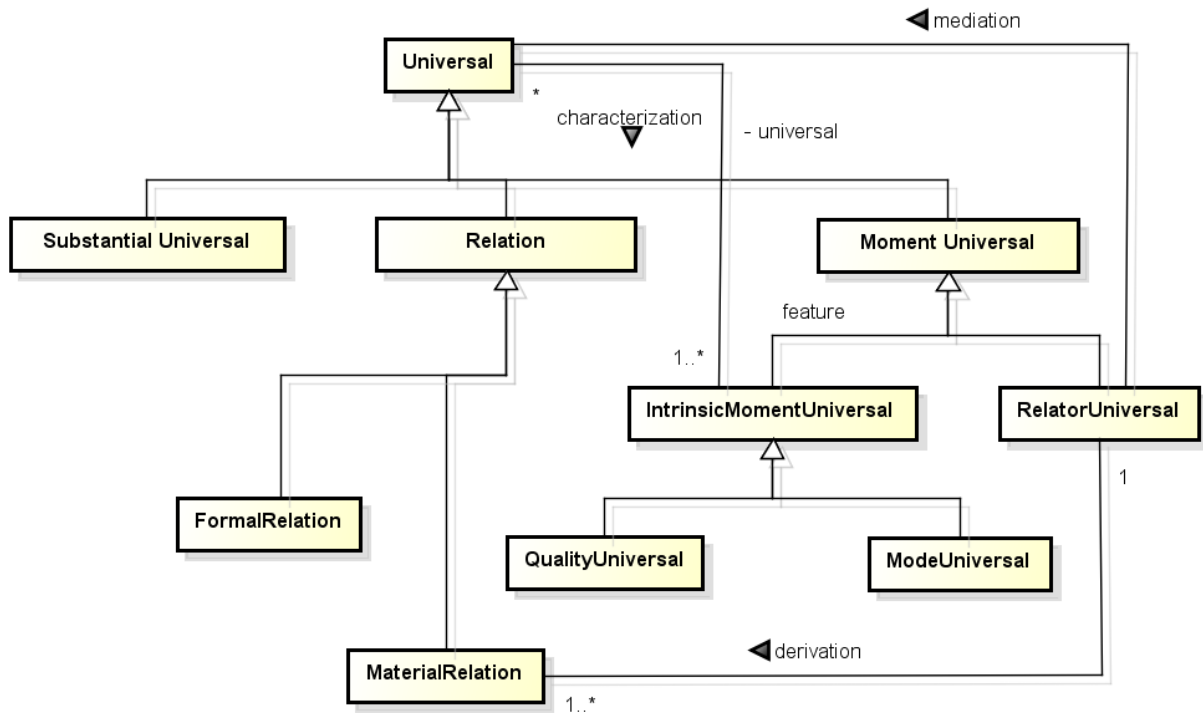


Figura 4-6. Tipologia da classe Moment Universal

A dependência existencial é utilizada para diferenciar entre *moments* intrínsecos e relacionais. A classe “*IntrinsicMomentUniversal*” representa propriedades que dependem apenas de um único indivíduo, enquanto que a classe “*RelatorUniversal*” representa as propriedades que dependem de vários indivíduos.

A classe “*IntrinsicMomentUniversal*” é especializada nas subclasses: “*QualityUniversal*”, relacionada a uma estrutura de qualidade, como, por exemplo, cor, peso, altura) e “*ModeUniversal*”, não relacionada a uma estrutura de qualidade, como, por exemplo, pensamento, sintoma, intenção.

A classe “Relation” representa as categorias de relações que podem ocorrer entre *Moments* e *Substantial*. A classe “Relation” é dividida em duas categorias gerais: “*MaterialRelation*” e “*FormalRelation*”.

A classe “*MaterialRelation*” tem uma estrutura material própria e suas relações são mediadas por um *Relator*. *Relators (moments)* são indivíduos com o poder de conectar entidades. Um exemplo pode ser observado na Figura 4-7, no qual o *Relator* “*Treatment*” conecta um “*Patient*” a um “*MedicalUnit*”. Uma relação *Material* é representada pelo estereótipo <<material>> (UML é a classe básica *Association*).

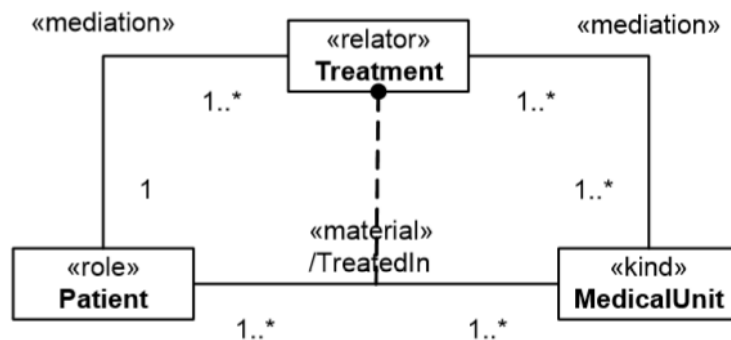


Figura 4-7. Representação de um Relator Universal

Já a classe “*FormalRelation*” representa relações entre duas ou mais entidades diretas sem mediação individual, como, por exemplo, “5 é maior do que 3”, “este dia faz parte deste mês”. Relações do tipo: herança, associações e dependência relacional, também são relações do tipo *Formal Relation*.

A relação *Formal* pode ser dividida em relações formais básicas e relações formais comparativas. Para as relações formais básicas são propostas na OntoUML três categorias: *Characterization*, *Mediation* e *Derivation*, as quais são especializadas de uma classe denominada *Dependence relationship* (Relação de Dependência), ou seja, são relações que dependem de outros indivíduos. Estes conceitos não têm representação correspondente no metamodelo da UML.

A relação *Formal* denominada *Characterization* ocorre entre um *mode universal* e um *universal* e não há propriedade opcional nesta relação. Uma relação *Formal Mediation* ocorre entre um *Relator Universal* e um *Endurant Universal*. Uma

relação *Derivation* é uma relação entre um *Material* e um *Relator Universal* de que esta relação é derivada.

Guizzardi ao longo de sua tese avalia várias deficiências das linguagens conceituais e, baseada nelas, os construtos são propostos de maneira a obter o compromisso ontológico da linguagem. Uma das deficiências discutidas na tese é a representação de relações todo-parte (na UML conhecidas como Agregação e Composição). Para estes tipos de relação são propostas quatro categorias: a) *subQuantityOf*, relaciona indivíduos que são *quantities*; b) *subCollectionOf*, relaciona indivíduos que são *collectives*; c) *memberOf*, relaciona indivíduos que são *functional complexes* ou *collectives* a indivíduos que são *collectives*; e d) *componentOf*, relaciona indivíduos que são *functional complexes*. A Figura 4-8 sumariza a hierarquia dos elementos derivados dos conceitos *Moments* e *Relation* e, a partir dos mesmos, são propostos os construtos OntoUML correspondentes, representados em cinza.

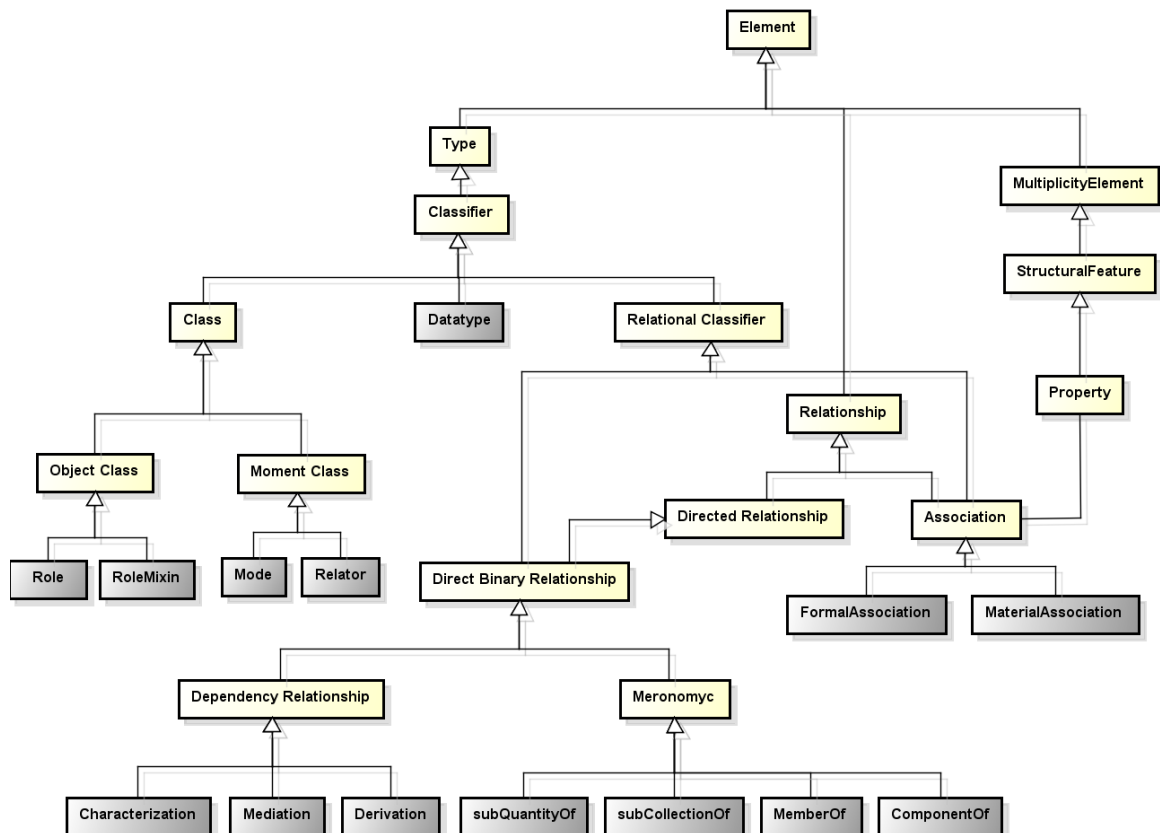
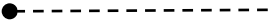




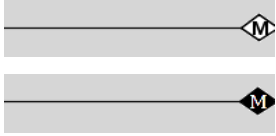
Figura 4-8. Fragmento do metamodelo UML para representação de categorias da OntoUML

O Quadro 4-2 resume os principais estereótipos utilizados na OntoUML referentes aos *Moments Universals*.

Quadro 4-2. Estereótipos da OntoUML referentes aos Moments Universals

Estereótipo	Descrição
<div style="border: 1px solid black; padding: 5px; text-align: center;"> <<role>> </div>	<p>Refinamento do estereótipo <<role>> apresentado no Quadro 4-1.</p> <p>Cada <<role>> deve estar conectado a um <i>association end</i> de uma relação <<relation>></p>
<div style="border: 1px solid black; padding: 5px; text-align: center;"> <<roleMixin>> </div>	<p>Refinamento do estereótipo <<roleMixin>> apresentado no Quadro 4-1.</p> <p>Cada <<roleMixin> deve estar conectado a um <i>association end</i> de uma relação <<mediation>></p>
<div style="border: 1px solid black; padding: 5px; text-align: center;"> <<mode>> </div>	<p>Um <<mode>> é um <i>Moment Universal</i> intrínseco. Cada instância de um <i>mode universal</i> é existencialmente dependente de exatamente uma entidade. Ex: habilidade, crenças, sintomas.</p> <p>Cada <<mode>> deve estar direta ou indiretamente conectado a um <i>association end</i> de pelo menos uma relação <<characterization>>.</p>
<div style="border: 1px solid black; padding: 5px; text-align: center;"> <<relator>> </div>	<p>Um <<relator>> é um <i>Moment Universal</i> relacional. Cada instância de um <i>mode universal</i> é existencialmente dependente de pelo menos duas entidades distintas. <i>Relators</i> são instâncias de propriedades relacionais, como por exemplo, casamento, compra, matrícula.</p> <p>Cada <<relator>> deve estar direta ou indiretamente conectado a um <i>association end</i> de pelo menos uma relação <<mediation>>.</p>
<div style="border: 1px solid black; padding: 5px; text-align: center;"> <<mediation>> </div>	<p>Uma relação <<mediation>> é uma relação formal que ocorre entre um <i>relator universal</i> e um <i>endurant universal</i> que faz a mediação. Ex: o universal Casamento media o <i>role universal</i> Marido e Esposa, o universal Matricula media o Estudante e a Universidade.</p> <p>Uma associação estereotipada com <<mediation>> deve ter na sua <i>association end</i> origem uma classe estereotipa com <<relator>>.</p> <p>As associações <<mediation>> são sempre binárias.</p>
<div style="border: 1px solid black; padding: 5px; text-align: center;"> <<characterization>> </div>	<p>Uma relação <<characterization>> é uma relação formal que ocorre entre um <i>mode universal</i> e um <i>endurant universal</i> que ele caracteriza. Ex: o universal Capacidade que caracteriza o universal Agente.</p> <p>Uma associação estereotipada com <<characterization>> deve ter na sua <i>association end</i> origem uma classe estereotipa com <<mode>>.</p> <p>A <i>association end</i> conectada para caracterizar um universal deve ter</p>

	<p>cardinalidade exatamente igual a 1.</p> <p>As associações <<characterization>> são sempre binárias.</p>
<p><i>Derivation Relation</i></p> 	<p>Uma relação <i>Derivation Relation</i> representa uma relação formal de uma derivação que ocorre entre um <i>material relation</i> e um <i>relator universal</i> de onde esta relação é derivada. Ex: casado-com, o qual é derivado de um <i>relator universal</i> Casamento.</p> <p>Um <i>Derivation Relation</i> deve ter uma de suas <i>association end</i> conectadas a um <i>relator universal</i> e outra conectada a <i>material relation</i>. Associações derivadas são sempre binárias.</p>
<p><<material>></p> <hr/>	<p>Uma associação <<material>> representa um <i>relator material</i>, ou seja, uma relação universal a qual é induzida por um <i>relator universal</i>. Ex: Estudante <u>estuda na</u> Universidade, Paciente é <u>tratado na</u> Unidade Médica.</p> <p>Cada <<material>> deve estar conectado a exatamente um <i>association end</i> de um <i>Derivation Relation</i>.</p>
<p><<formal>></p> <hr/>	<p>Uma associação <<formal>> representa um <i>Formal Relation</i>, ou seja, uma relação de comparação derivada de propriedades intrínsecas das entidades relacionadas, ou uma relação interna. Ex: uma Pessoa é <u>mais velha do</u> que Pessoa, um Átomo é <u>mais pesado do que</u> outro Atomo.</p>
	<p>componentOf é uma relação de especialização da classe abstrata Meronymic, que representa as propriedades de todas as relações meronímicas (todo-parte). Um componentOf é uma relação todo-parte entre dois complexos funcionais (kind ou subkind). Ex: uma Mão faz parte de um Braço, um Motor faz parte de um carro. A relação pode ser compartilhável (losango branco) ou não compartilhável (losango preto).</p>
	<p>subQuantityOf é uma relação todo-parte entre dois <i>quantities</i>. Ex: Álcool faz parte do Vinho, Plasma faz parte do Sangue.</p> <p>A relação é sempre <i>non-shareable</i> e essencial.</p> <p>Ambas as <i>association end</i> desta relação devem representar universais cujas instâncias são <i>quantities</i>. Cardinalidade máxima igual a 1.</p>
	<p>subCollectionOf é uma relação todo-parte entre dois <i>collectives</i> Ex: conjunto de facas em um conjunto de talheres é parte de um conjunto de talheres.</p> <p>A relação pode ser <i>shareable</i> ou <i>non-shareable</i>. O losango preenchido corresponde a uma relação <i>non-shareable</i>.</p>

	<p>Ambas as <i>association end</i> desta relação devem representar universais cujas instâncias são <i>collectives</i>.</p> <p>Cardinalidade máxima igual a 1.</p>
	<p>memberOf é uma relação todo-parte entre um <i>complex</i> ou um <i>collective</i> e um <i>collective</i>. Ex: uma árvore é parte de uma floresta, uma faca é parte de um conjunto de talheres.</p> <p>A relação pode ser <i>shareable</i> ou <i>non-shareable</i>. O losango preenchido corresponde a uma relação <i>non-shareable</i>.</p> <p>O classificador conectado na <i>association end</i> relacionado ao indivíduo todo deve ser um universal cujas instâncias são <i>collectives</i>. O classificado conectado a <i>association end</i> relacionado a parte pode ser instâncias de <i>collectives</i> ou <i>functional complexes</i>.</p>
<<datatype>>	<p>Datatype é um classificador, semelhante a uma classe, mas cujas instâncias são valores e não objetos</p>

Na área de modelagem conceitual, outras linguagens e metamodelos são mais populares nas empresas do que a OntoUML, como por exemplo, o meta modelo ER (Entidade Relacionamento) e a UML. O meta modelo ER é um dos mais utilizados nas empresas. A razão da sua popularidade, no entanto, é exatamente a sua principal fraqueza: o meta modelo é muito simples, o que torna fácil a sua utilização pelos modeladores, porém, não apresenta alta representatividade (CASTRO, 2010). A UML também se apresenta como uma linguagem popular nas empresas, porém padece do mesmo problema de expressividade. Guizzardi (2005) discute exaustivamente diversos problemas de sobrecarga, redundância, falta de clareza na UML e em outras linguagens de modelagem.

4.4 Considerações sobre o capítulo

Neste capítulo a OntoUML foi apresentada como a principal linguagem para a construção destes modelos conceituais ontológicos. Os construtos da OntoUML foram apresentados em detalhes assim como as limitações das linguagens tradicionais para modelagem conceitual. Após o aprofundamento do referencial teórico desta tese, o próximo capítulo apresenta a abordagem metodológica utilizada para a condução desta pesquisa.

CAPÍTULO 5 - ESTRUTURAÇÃO DA PESQUISA

Este capítulo tem o objetivo de apresentar a abordagem metodológica aplicada para o desenvolvimento desta pesquisa. Para tanto, se faz necessário abordar algumas referências para estabelecer os conceitos necessários. Na sequência, foi caracterizada a forma de trabalho e definida a estratégia e as etapas utilizadas pelo pesquisador para estruturar seu trabalho e alcançar os objetivos previamente definidos.

5.1 Caracterização da pesquisa

Quanto aos objetivos, esta pesquisa foi classificada como exploratória, pois pretende explorar o uso de modelos conceituais em OntoUML para apoiar o processo de derivação de requisitos funcionais de domínio. Quanto ao procedimento técnico, nenhuma das classificações previstas em Gil (2002) foi considerada inteiramente adequada a este trabalho. Desta maneira, esta pesquisa foi estruturada combinando diversos procedimentos técnicos, os quais serão detalhados de acordo com a etapa da pesquisa apresentada.

5.2 Estratégia da pesquisa

Como estratégia desta pesquisa, foi definida uma estrutura própria combinando vários métodos e procedimentos técnicos para atingir os objetivos iniciais. Como pode ser visto na Figura 5-1 esta estrutura é dividida em cinco etapas principais: Exploração do Objeto, Avaliação da Expressividade da OntoUML, Proposta de Método para a Derivação de Requisitos Funcionais de Domínio, Desenvolvimento de Ambiente para a Construção do Modelo Conceitual OntoUML e por fim a Avaliação da Abordagem. Todas as etapas são detalhadas a seguir.

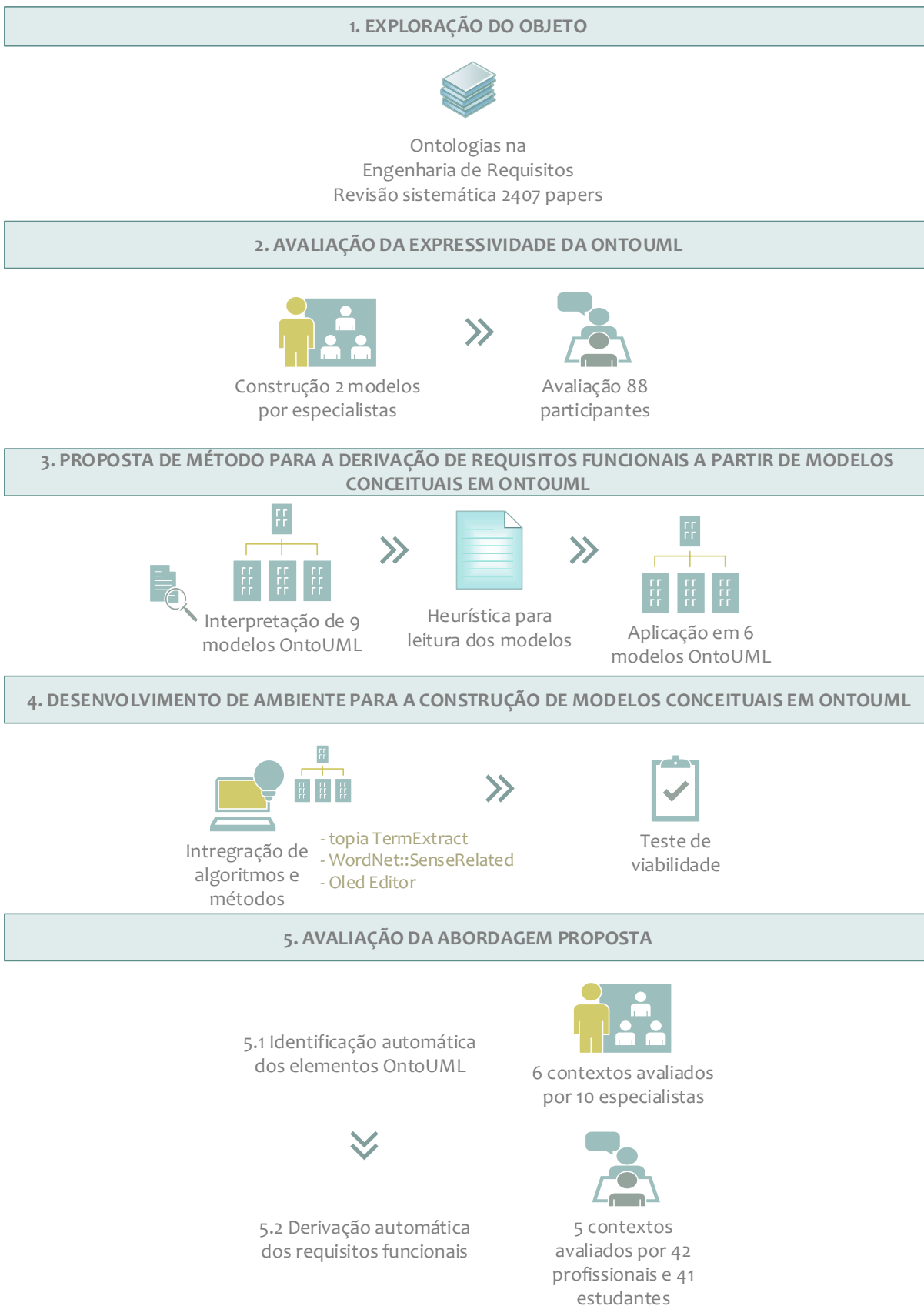


Figura 5-1. Estrutura da pesquisa

5.2.1 Exploração do objeto

Nesta fase foi usado o método denominado revisão sistemática de literatura, que é um tipo de estudo secundário (KITCHENHAM, 2004). Estudos secundários visam identificar, avaliar e interpretar todos os resultados relevantes a um determinado tópico de pesquisa, fenômeno de interesse ou questão de pesquisa (KITCHENHAM, 2004). Os resultados obtidos por diversos estudos primários correlatos atuam como fonte de informação a ser investigada por estudos secundários. A precisão e a confiabilidade proporcionadas pelos estudos secundários contribuem para a melhoria e para o direcionamento de novos tópicos de pesquisa, a serem investigados por estudos primários.

Seguindo as etapas propostas por Kitchenham (KITCHENHAM, 2004) foi realizada a revisão sistemática, de forma a elucidar o campo de estudo e identificar lacunas para contribuição científica. A revisão é apresentada no Capítulo 3 e teve o objetivo de responder as questões:

1. Quais são as atividades da ERS nas quais as ontologias têm sido aplicadas?
2. Quais são as funções das ontologias na ERS?
3. Quais são as linguagens usadas pelas ontologias?
4. Quais são os domínios do conhecimento representados?
5. Quais são as contribuições das ontologias para a ERS?

A partir de análise inicial de 2.407 artigos, a revisão sistemática proveu uma visão geral do campo de estudo, dando suporte à definição dos objetivos e questões de pesquisa no Capítulo 1.

5.2.2 Avaliação da expressividade da OntoUML

O resultado da revisão sistemática ressaltou a importância do uso de modelos conceituais ontologicamente fundamentados como instrumento para aprimorar a expressividade do conhecimento de domínio. A revisão sistemática também mostrou que a linguagem utilizada para a representação do modelo conceitual contribui para a qualidade da comunicação. Com esta motivação, a linguagem OntoUML foi a escolhida para a representação de modelos conceituais no escopo desta tese. Embora Guizzardi (GUIZZARDI, 2005) tenha ilustrado em sua tese situações em que a OntoUML possibilita melhor representação evitando sobrecarga e redundância,

considerou-se importante realizar uma avaliação dentro de um contexto de sistemas de informação para verificar a expressividade da OntoUML em relação a UML.

O experimento foi conduzido em dois grupos com conhecimento em modelagem conceitual, sendo um grupo composto por 8 profissionais da área de computação e outro grupo formado por 80 estudantes da área de computação. Dois modelos conceituais de um mesmo domínio foram construídos por 3 especialistas na linguagem OntoUML e 3 especialistas na linguagem UML.

Estes modelos foram apresentados aos participantes junto com um conjunto de 12 afirmações relacionadas ao domínio (APÊNDICE A). Os participantes deveriam selecionar em qual dos dois modelos conceituais (OntoUML e UML) a afirmação estava representada de forma mais clara. Detalhes da avaliação e os resultados são apresentados no Capítulo 6.

5.2.3 Proposta de método para a derivação de requisitos funcionais a partir de modelos conceituais em OntoUML

A avaliação da expressividade da ONTOUML demonstrou diversas situações onde a OntoUML permite representações mais claras relacionados ao domínio. Este resultado reforçou a motivação desta tese de que o modelo conceitual representado em OntoUML poderia ser um instrumento relevante para apoiar a derivação de requisitos funcionais de domínio. No entanto, não havia nenhum método formal que orientasse a derivação de requisitos funcionais de domínio. Portanto, um método foi proposto com este objetivo.

Foram estudados 9 modelos conceituais representados em OntoUML e, por meio deles, foi definida uma heurística com o objetivo de fazer a leitura e a transcrição textual destes modelos. A heurística deveria percorrer todo o modelo, sem repetição e a transcrição deveria usar linguagem natural para representar possíveis requisitos funcionais de domínio. A heurística proposta foi aplicada em 6 dos 9 modelos inicialmente avaliados. O Capítulo 7 apresenta o método proposto e os resultados correspondentes.

5.2.4 Desenvolvimento de ambiente para a construção de modelo conceitual em OntoUML

Embora o modelo conceitual representado em OntoUML tenha se apresentado como um instrumento relevante para aprimorar a representação de um

domínio do conhecimento, a sua construção não é uma tarefa trivial. Por si só, a construção de um modelo conceitual, em qualquer linguagem, requer habilidade de abstração e visão global. Estas habilidades são desenvolvidas pelos modeladores com a experiência. Além disso, a disponibilidade de um conjunto maior de construtos, que é o caso da OntoUML, aumenta a complexidade de sua construção. Considerando estas dificuldades, Guizzardi (GUIZZARDI, 2005) enfatiza a necessidade de desenvolver métodos e ferramentas que apoiem o processo de construção do modelo conceitual. Sendo assim, foi desenvolvido um ambiente computacional com o objetivo de apoiar a construção semiautomática do modelo conceitual em OntoUML. Para o desenvolvimento do ambiente algumas etapas foram executadas, sendo as principais listadas a seguir.

5.2.4.1 Definição das funcionalidades do ambiente

Nesta etapa foram definidas quais seriam as atividades a serem executadas por meio do ambiente de maneira a garantir a construção do modelo conceitual a partir de um texto e ter como saída uma lista de requisitos. Foram definidas as atividades com seus respectivos fluxos de execução.

5.2.4.2 Desenvolvimento da arquitetura do ambiente

Nesta etapa foram definidos os algoritmos, métodos e ferramentas necessários para permitir que todas as funcionalidades previstas fossem executadas automaticamente. A seguir alguns detalhes dos instrumentos definidos e integrados ao ambiente:

- Algoritmo para a identificação de termos relevantes: foi realizada uma pesquisa exploratória para definir os critérios e a seleção do algoritmo mais adequado para esta tarefa;
- Algoritmo para desambiguação de termos: foi definido e integrado um algoritmo que realizasse a desambiguação de termos usando a base WordNet;
- Método para a identificação do construto OntoUML: foi integrado o método proposto por Leão et al. (LEÃO et al., 2013) e adicionado novas regras para aprimorar o método;
- Ferramenta para a construção do modelo conceitual em OntoUML: foi integrado o editor OntoUML OLED (OLED, 2014) ao ambiente; e

- Método para a derivação de requisitos de domínio: o método proposto na Seção 5.2.3 foi formalizado e integrado ao ambiente.

5.2.4.3 *Teste de viabilidade*

Foi realizado um teste de viabilidade com o objetivo de avaliar a operacionalização do ambiente, com relação a construção semiautomática do modelo conceitual. Neste teste, foi usado como entrada um texto em inglês referente ao domínio de roteiro de ônibus e a saída foi a versão inicial do modelo conceitual em OntoUML referente a este domínio. Foram usadas as métricas *precision* e *recall* para avaliar o desempenho do algoritmo para seleção dos termos relevantes e o conjunto de conceitos identificados automaticamente pelo ambiente.

O teste de viabilidade apontou que é possível semiautomatizar algumas tarefas e outras requerem intervenção dos especialistas. No entanto, resultados mais expressivos seriam obtidos apenas por meio da realização de experimentos com especialistas e usuários finais. O Capítulo 8 apresenta o fluxo das atividades implementadas no ambiente, a arquitetura e o resultado do teste de viabilidade.

5.2.5 **Avaliação da abordagem proposta**

Com os métodos implementados e internamente validados, foram realizadas duas avaliações. A primeira avaliação foi conduzida para verificar o método de identificação automática dos elementos em um modelo conceitual OntoUML e a segunda avaliação foi conduzida para verificar o método de derivação de requisitos funcionais de domínio a partir do modelo conceitual OntoUML.

5.2.5.1 *Avaliação da identificação automática dos elementos de modelos conceituais em OntoUML*

O objetivo desta avaliação foi verificar a assertividade do método para a identificação automática dos elementos em modelo conceitual OntoUML. Seis descrições de domínio foram selecionadas a partir de publicações científicas que discutiam o uso de processamento de linguagem natural, com o objetivo de construir modelo conceitual para esta avaliação. Os textos foram processados seguindo as atividades descritas no Capítulo 8. Ao final do processamento automático, para cada descrição de domínio, uma lista dos termos identificados e seus respectivos construtos foram gerados. Foram elaborados 2 instrumentos (APÊNDICE B), cada um com o resultado de 3 descrições de domínio. Cada instrumento foi enviado para

16 especialistas. A lista inicial dos especialistas foi obtida por meio da seleção de publicações científicas relacionadas a OntoUML. Para apoiar a avaliação, 4 métricas foram estabelecidas. Porém, para que fosse possível chegar a estas métricas, 5 medidas foram coletadas no experimento. O Quadro 5-1 apresenta o identificador, o nome e a descrição das 5 medidas definidas. Já o Quadro 5-2 apresenta os dados das 4 métricas definidas no experimento e como elas foram calculadas utilizando as medidas do Quadro 5-1.

ID medida	Nome	Descrição
a	Relevantes Método	Soma dos termos identificados como relevantes pelo método
b	Relevantes especialistas	Considerando os termos identificados em (a), soma dos termos indicados como relevantes pelos especialistas. Foram considerados apenas os termos que obtiveram consenso entre os especialistas (3 indicações)
c	Desambiguados corretamente	Considerando os termos em (b), soma dos termos desambiguados corretamente pelo algoritmo TargetWord, avaliação feita manualmente por meio do WordNet
d	Construto OntoUML especialistas	Considerando os termos identificados em (b), soma dos termos que obtiveram consenso entre os especialistas com relação a indicação do construto OntoUML
e	Acerto método	Considerando os termos identificados em (d), soma dos termos em que os especialistas indicaram o mesmo construto OntoUML indicado pelo método

Quadro 5-1. Medidas coletadas no experimento

ID métrica	Nome	Descrição	Medida (Cálculo)
A	Relevantes corretos	Percentual de termos relevantes identificados corretamente pelo método	$b/a*100$
B	Desambiguados corretos	Percentual de termos relevantes identificados corretamente pelo método e desambiguados corretamente	$c/b*100$
C	Consenso construto OntoUML	Percentual de consenso entre os especialistas na indicação do construto OntoUML	$d/b*100$
D	Acerto método	Percentual de acerto do método proposto para a identificação de termos relevantes e seu respectivo construto OntoUML	$e/d*100$

Quadro 5-2. Métricas utilizadas no experimento

O Capítulo 9 apresenta em detalhes o método e o resultado do experimento.

5.2.5.2 Avaliação da derivação de requisitos funcionais de domínio a partir de modelos conceituais em OntoUML

A segunda avaliação teve o objetivo de verificar a completude de uma lista de requisitos de domínio gerada automaticamente por meio do ambiente computacional proposto, a partir de um modelo conceitual OntoUML. Foram usados 5 dos 6 textos aplicados na avaliação anterior (avaliação da identificação automática dos elementos). Conforme mencionado, o método proposto para apoiar a construção de modelo conceitual não identifica as relações entre os conceitos. Portanto, os modelos gerados na avaliação anterior foram completados manualmente baseado nos feedbacks dos especialistas em OntoUML para serem utilizados nesta avaliação.

Para esta avaliação foram elaborados 3 instrumentos (APÊNDICE C), sendo que os instrumentos 1 e 2 possuem 2 descrições de domínio distintos e o instrumento 3 apenas 1 descrição de domínio distinto. O instrumento 3 foi elaborado com apenas 1 descrição por se tratar do texto mais extenso entre os selecionados. Junto com o texto foi enviada a lista de requisitos de domínio gerada automaticamente pelo método a partir do modelo conceitual OntoUML. O objetivo geral foi identificar a cobertura do método com relação a geração de requisitos funcionais de domínio considerando uma descrição de domínio.

Os instrumentos foram aplicados em dois grupos distintos: o primeiro, denominado profissionais, participantes graduados e com experiência na atividade de levantamento de requisitos, e o segundo denominado estudantes, participantes cursando graduação em computação que já cursaram disciplinas relacionadas a atividade de levantamento de requisitos. Estes dois grupos foram estabelecidos para verificar se a experiência seria um fator influenciador nos resultados. O Capítulo 9 apresenta em detalhes o método e o resultado da avaliação.

5.3 Considerações sobre o capítulo

Este capítulo apresentou a estruturação da pesquisa conduzida no escopo desta tese. Foram apresentadas as cinco principais etapas executadas para chegar aos resultados finais da pesquisa. Em cada etapa procurou-se dar uma visão geral das motivações, dos métodos e dos critérios utilizados. Os resultados correspondentes são apresentados em detalhes nos capítulos seguintes.

CAPÍTULO 6 - AVALIAÇÃO DA EXPRESSIVIDADE DA ONTOUML

Embora em sua proposta Guizzardi (GUIZZARDI, 2005) discuta profundamente as situações em que a OntoUML é mais expressiva do que as demais linguagens para a construção de modelos conceituais, considerou-se importante conduzir uma avaliação relacionada a expressividade destes modelos quando usadas as linguagens OntoUML e UML. A avaliação ocorreu em um contexto aplicado e relacionado a Sistemas de Informação, pois não foram encontradas avaliações com este propósito. Este capítulo apresenta o método e os resultados desta avaliação. Os resultados desta avaliação também estão publicados em (VALASKI et al., 2016b).

A Figura 6-1 situa a etapa da pesquisa correspondente a este capítulo.



Figura 6-1. 2ª etapa da pesquisa

6.1 Fases da avaliação

Esta seção descreve as fases da avaliação conduzida para avaliar a expressividade das linguagens OntoUML e UML.

6.1.1 Seleção do domínio

A primeira fase da avaliação consistiu na escolha do domínio para a construção do modelo conceitual. O objetivo foi selecionar um domínio não comum, ou seja, não amplamente conhecido (universidade, biblioteca e locadora), porém com um escopo reduzido para não inviabilizar a avaliação. O uso de um domínio não comum poderia gerar mais discussões a respeito dos conceitos e relações existentes. De acordo com estes critérios, a especificação de requisitos de software no contexto da procuração eletrônica foi obtida com especialistas do domínio. Baseada nesta especificação, uma descrição das principais funcionalidades do domínio foi realizada. O Quadro 6-1 apresenta a descrição do domínio usada para a construção dos modelos conceituais.

Quadro 6-1. Descrição do domínio

Descrição
<p>Somente o representante da organização pode outorgar uma procuração eletrônica. Uma organização pode ter um ou mais representantes. Deve ser permitido ao outorgante indicar um usuário ativo da base da Receita-PR a condição de outorgado. Deverá existir apenas um outorgado por procuração. Deverá ser permitida apenas uma procuração com o mesmo outorgado e outorgante. Apenas organizações com registro na base de dados do ICMS podem ser relacionadas à uma procuração. Devem ser apresentados os serviços a serem outorgados. Devem ser listadas as organizações (a qual o outorgante é representante) a serem outorgadas. Permitir o outorgante revogar uma procuração.</p>

6.1.2 Construção do modelo conceitual em OntoUML

Baseado no escopo definido no Quadro 6-1, especialistas em OntoUML construíram o respectivo modelo conceitual usando a linguagem OntoUML. Como a OntoUML ainda não é uma linguagem amplamente utilizada nas empresas, são encontrados poucos especialistas. O *Conceptual Modeling Research Group* (NEMO), liderado pelo professor Giancarlo Guizzardi, autor da OntoUML, é um grupo que desenvolve intensamente pesquisas usando a OntoUML. Portanto, este grupo foi selecionado para participar desta fase da avaliação. Um e-mail foi enviado ao grupo NEMO com a descrição do domínio e foi solicitada a construção do respectivo modelo conceitual. A construção foi realizada colaborativamente entre

três membros do grupo. E-mails foram trocados para esclarecer dúvidas dos especialistas quanto ao domínio.

6.1.3 Construção do modelo conceitual em UML

Para a construção do modelo conceitual em UML, três especialistas foram selecionados. Todos os selecionados têm níveis consolidados de conhecimento em UML, com mais de 10 anos de experiência profissional na indústria e na academia. A descrição do domínio também foi enviada por e-mail solicitando a cada especialista construir individualmente o respectivo modelo conceitual. E-mails foram trocados para esclarecer dúvidas dos especialistas quanto ao domínio.

6.1.4 Avaliação da expressividade dos modelos

O objetivo desta fase foi avaliar a expressividade dos dois modelos conceituais construídos pelos especialistas (OntoUML e UML). Doze sentenças foram manualmente extraídas a partir da descrição do domínio. Usando estas sentenças, um instrumento foi preparado para avaliar em qual dos modelos a sentença tinha uma representação mais clara. Ou ainda, se ambos os modelos apresentavam o mesmo nível de clareza. O instrumento criado para a avaliação, está incluído no Apêndice A.

Após a elaboração do instrumento, o perfil dos participantes da avaliação foi definido. Dois grupos distintos foram selecionados: o primeiro composto por oito profissionais formados na área da computação e com experiência em modelagem UML e o segundo grupo composto por oitenta estudantes da graduação em computação. O experimento foi realizado apenas com os estudantes que já haviam concluído disciplinas que abordassem a UML. Nenhum dos dois grupos tinham conhecimento prévio referente a OntoUML. O experimento foi primeiramente conduzido com o grupo de profissionais para validar o instrumento. Baseada nesta experiência, algumas melhorias foram aplicadas no instrumento com o grupo de estudantes. Uma das melhorias foi a criação de duas versões do instrumento. Na primeira versão a UML aparecia sempre como a primeira opção e na segunda versão a OntoUML aparecia sempre como a primeira opção. Esta adequação foi necessária para evitar o viés da ordem das opções. As versões foram distribuídas alternadamente.

6.2 Resultados e discussão

Nesta seção, são apresentados e discutidos os resultados da avaliação. Primeiramente, são apresentados os resultados relacionados à construção dos modelos conceituais pelos especialistas e, posteriormente, são apresentados os resultados relacionados a avaliação da expressividade dos modelos por profissionais e estudantes.

6.2.1 Modelo conceitual em OntoUML

A Figura 6-2 apresenta o modelo conceitual construído pelos especialistas em OntoUML. Para atingir a versão final foram realizadas quatro rodadas de esclarecimento de dúvidas quanto à descrição do domínio.

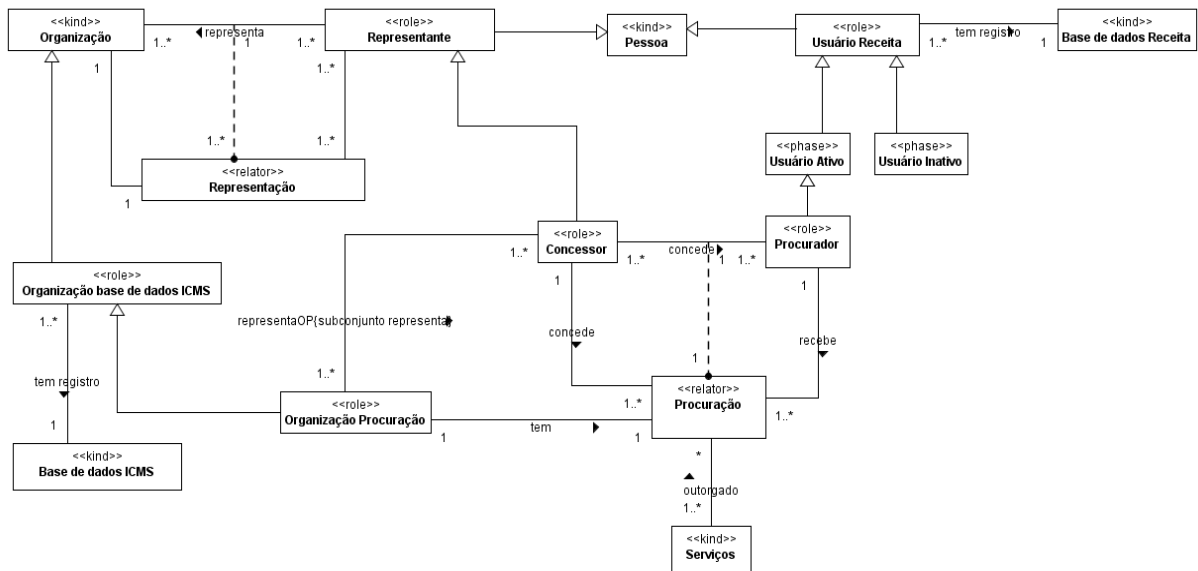


Figura 6-2. Modelo conceitual em OntoUML

Os especialistas em OntoUML revelaram que a riqueza dos construtos da linguagem gerou diversas questões relativas ao escopo. Os especialistas também observaram que muita informação que está implícita na descrição precisa se tornar explícita ao usar a OntoUML. Em todas as rodadas, os especialistas indicaram informações que precisavam ser incluídas na descrição do escopo para tornar a construção do modelo mais completa. Segundo os especialistas, muita informação estava implícita.

No experimento foi observado que o alto grau de formalidade e consistência no uso da linguagem OntoUML gera diversas questões que talvez não seriam geradas se outras linguagens fossem usadas. Este feedback reforça a crença de

que modelos conceituais representados em OntoUML tenham um impacto positivo no apoio a eliciação de requisitos de software. Também é importante pontuar que a maior expressividade da linguagem gera maior complexidade de uso, uma vez que, o maior número de construtos requer maior aprendizado e cuidado.

6.2.2 Modelo conceitual em UML

A Figura 6-3 representa o modelo conceitual final construído pelos especialistas em UML. O comportamento dos especialistas em UML com relação às dúvidas sobre a descrição do domínio foram diferentes do comportamento dos especialistas em OntoUML. O especialista 1 entregou a versão sem qualquer dúvida. O especialista 2 solicitou apenas de uma rodada de dúvidas e entregou a versão. Finalmente, o especialista 3 solicitou duas rodadas de dúvidas antes da entrega da versão final.

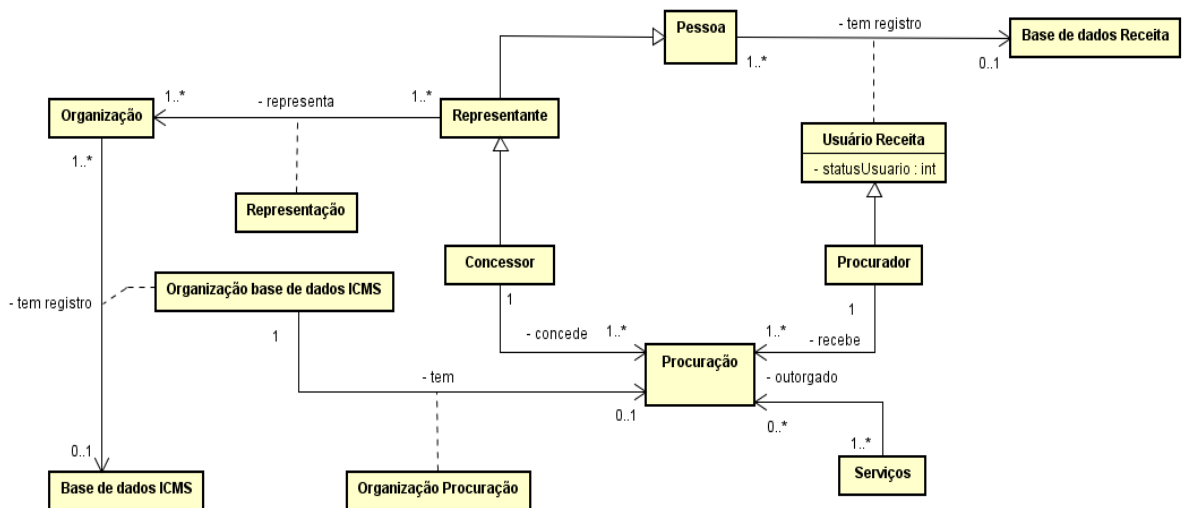


Figura 6-3. Modelo conceitual em UML

As três versões entregues diferiram consideravelmente. Algumas das razões podem ser: o baixo grau de formalismo da linguagem UML permite representações distintas do mesmo contexto; e a falta de restrições semânticas não encoraja o questionamento durante a construção. As versões entregues focaram na persistência de dados necessária para o funcionamento do software em vez de focar na representação de conceitos do domínio.

A versão entregue pelo especialista 3 foi a mais próxima da representação do escopo definido no experimento. Em um encontro presencial entre os especialistas

concluiu-se uma versão final do modelo. Com os dois modelos conceituais (OntoUML e UML) construídos pelos especialistas, a avaliação da expressividade entre ambos foi realizada. Os resultados são apresentados a seguir.

6.2.3 Avaliação da expressividade dos modelos conceituais construídos

Primeiramente são apresentados os resultados do experimento piloto realizado com o grupo de profissionais. A Tabela 6-1 apresenta uma visão geral destes resultados. Considerando que oito profissionais avaliaram doze sentenças, noventa e seis escolhas foram coletadas. Entre estas escolhas, doze (13%) indicaram a UML como a linguagem mais expressiva, quarenta indicaram a OntoUML (42%) e quarenta e quatro indicam que ambas as linguagens possuem o mesmo nível de clareza.

Tabela 6-1. Resultado consolidado do grupo de profissionais

Linguagem	Número de escolhas	Percentual
UML	12	13%
OntoUML	40	42%
Ambas	44	46%
Total	96	100%

Baseado nestes resultados, foram identificadas as sentenças nas quais as linguagens apresentavam o mesmo nível de clareza e as sentenças nas quais a OntoUML apresentava nível de clareza superior a UML. Portanto, uma nova análise foi feita por sentença e os resultados são apresentados na Figura 6-4.

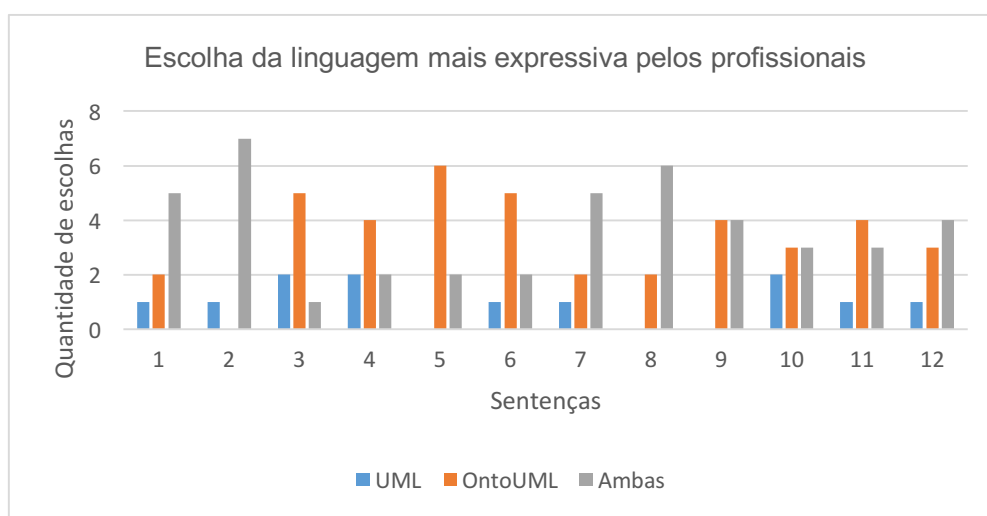


Figura 6-4. Quantidade de escolhas pelos profissionais para cada uma das doze sentenças

A Figura 6-4 mostra que para as sentenças: 1, 2, 7, 8 e 12, ambas as linguagens apresentaram o mesmo nível de clareza. Para as sentenças: 3, 4, 5, 6, e 11, a OntoUML apresentou maior nível de clareza. Não houve nenhuma sentença indicando a UML com maior nível de clareza do que a OntoUML.

Considerando que os profissionais têm mais experiência prática com a modelagem do que os estudantes, os profissionais poderiam ter percepções diferentes dos estudantes. Por esta razão, o experimento considerou estes dois grupos distintos para identificar estas percepções.

A Tabela 6-2 apresenta uma visão geral dos resultados obtidos pelos estudantes. Oitenta estudantes avaliaram as doze sentenças, totalizando novecentos e sessenta escolhas. Entre as escolhas, cento e oitenta e uma (19%) indicaram a UML como a linguagem mais expressiva, quatrocentos e seis (42%) indicaram a OntoUML com a mais expressiva e trezentos e setenta e três (39%) indicaram que ambas as linguagens possuem o mesmo nível de clareza. Apesar dos estudantes terem menos experiência prática sobre modelagem, a percepção deles foi muito similar à percepção dos profissionais.

Os resultados também foram analisados individualmente pelos grupos de estudantes, uma vez que pertenciam a períodos, cursos e instituições distintas. A Tabela 6-3 apresenta estes resultados também indicando o curso, o período corrente do estudante e a quantidade de estudantes de cada grupo. Todos os grupos concordaram que a OntoUML foi a mais expressiva para a maioria das sentenças. A exceção foi o Grupo 3 no qual houve empate (45%). A maior diferença dos resultados foi entre os Grupos 1 e 5, indicando um percentual para a UML de 6% no Grupo 1 e 29% no Grupo 5. Esta diferença pode estar relacionada ao grau de conhecimento do estudante com relação a UML. No entanto, não é possível fazer afirmações a este respeito baseado apenas neste experimento.

Tabela 6-2. Resultado consolidado do grupo de estudantes

Linguagem	Número de escolhas	Percentual
UML	181	19%
OntoUML	406	42%
Ambas	373	39%
Total	960	100%

Tabela 6-3. Escolhas dos estudantes por classe

ID	Curso	Período	No. de estudantes	UML	OntoUML	Ambas
Grupo 1	Ciência da computação	5	12	6%	49%	45%
Grupo 2	Sistema de informação	7	12	22%	46%	32%
Grupo 3	Engenharia da computação	7	17	10%	45%	45%
Grupo 4	Sistema de informação	6	9	14%	45%	41%
Grupo 5	Tecnologia em análise e desenvolvimento de sistemas	5	30	29%	36%	35%

Assim como foi feito com os resultados dos profissionais, as sentenças foram avaliadas individualmente com relação aos resultados dos estudantes. Os resultados agrupados por sentença são apresentados na Figura 6-5. O resultado mostra que para as sentenças: 1, 2, 7, 9 e 12, ambas as linguagens exibiram o mesmo nível de clareza. Para as sentenças: 3, 4, 5, 6, 8, 10 e 11 a OntoUML apresentou maior nível de clareza. Nenhuma sentença indicou a UML com maior nível de clareza.

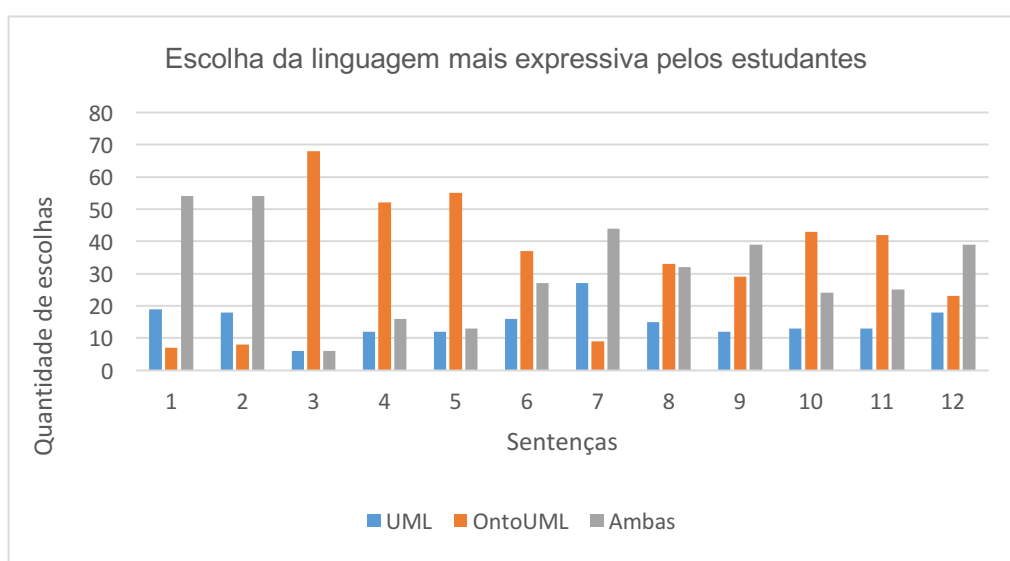


Figura 6-5. Quantidade de escolhas pelos estudantes para cada uma das doze sentenças

Assim, como ocorreu com os resultados obtidos pelos profissionais, estes resultados indicaram situações nas quais a OntoUML é mais expressiva e situações nas quais ambas as linguagens apresentaram o mesmo nível de clareza. Para evidenciar esta observação, os resultados foram agrupados por sentença e por cada grupo participante. Este agrupamento é apresentado no Quadro 6-2. O resultado revela consenso no que se refere a maior clareza da OntoUML para as sentenças: 3, 4, 5, 6, 10 e 11 (em cinza).

Quadro 6-2. Seleção da linguagem mais expressiva por sentença e por grupo

Sentença	Profissionais	Estudantes	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5
1	Ambas	Ambas	Ambas	Ambas	Ambas	Ambas	Ambas
2	Ambas	Ambas	Ambas	Ambas	Ambas	Ambas	Ambas
3	OntoUML	OntoUML	OntoUML	OntoUML	OntoUML	OntoUML	OntoUML
4	OntoUML	OntoUML	OntoUML	OntoUML	OntoUML	OntoUML	OntoUML
5	OntoUML	OntoUML	OntoUML	OntoUML	OntoUML	OntoUML	OntoUML
6	OntoUML	OntoUML	Ambas	OntoUML	Ambas	OntoUML	OntoUML
7	Ambas	Ambas	Ambas	Ambas	Ambas	Ambas	Ambas
8	Ambas	OntoUML	OntoUML	Ambas	OntoUML	Ambas	Ambas
9	Empate	Ambas	OntoUML	Ambas	Ambas	Ambas	Ambas
10	Empate	OntoUML	OntoUML	OntoUML	OntoUML	Ambas	OntoUML
11	OntoUML	OntoUML	OntoUML	OntoUML	OntoUML	Ambas	OntoUML
12	Ambas	Ambas	Ambas	OntoUML	Ambas	OntoUML	UML

Na linguagem OntoUML, o construto Role foi usado para representar as sentenças 4, 5, 6, 10 e 11. A OntoUML usa o construto Role para representar relações que são dependentes de um conceito Universal, o qual carrega o princípio de identidade e individualização. Nas sentenças avaliadas no modelo OntoUML, uma relação de especialização foi requerida. Na UML, devido à baixa restrição semântica, os conceitos para as mesmas sentenças são representados por meio de associações, as quais nem sempre deixam claras a origem do conceito, ao contrário das representações em OntoUML. Esta constatação fica mais clara quando se observa a sentença 9. Nesta sentença, as percepções dos participantes são as mesmas para ambas as linguagens, apesar do uso de um Role no modelo OntoUML. Isto ocorreu, porque na UML também foi utilizada uma especialização para representar o conceito, o que resultou no mesmo nível de clareza em ambos os modelos.

Na sentença 3 foi usado o construto Relator. Este construto permite maior clareza da multiplicidade das relações. Na UML foi usada uma relação associativa. As relações associativas da UML não permitem representações claras de determinadas relações. Por exemplo, a representação de que pode haver apenas uma relação entre o mesmo outorgante e outorgado não é possível na UML, embora um outorgante possa estar relacionado a vários outorgados e vice-versa. Guizzardi (GUIZZARDI, 2005) discute esta deficiência em diversas linguagens.

É possível que se os participantes conhecessem os construtos OntoUML, os resultados poderiam ser ainda mais favoráveis a escolha da OntoUML como a

linguagem com representação mais clara. Um exemplo, que não foi percebido pelos participantes, refere-se à representação da sentença 9. Em OntoUML é possível representar que não é suficiente o outorgado ser o representante de uma organização, mais que ele precisa ser o representante da organização que está associada à procuração.

6.3 Considerações sobre a avaliação

Como já discutido em capítulos anteriores, os modelos conceituais são considerados um importante instrumento para obter consenso e entendimento de um domínio do conhecimento. Por esta razão, modelos conceituais podem ser um aliado para apoiar o processo de elicitação de requisitos em domínios pouco conhecidos e/ou mais complexos. No entanto, a linguagem usada para a representação do modelo pode comprometer a sua expressividade. Embora a UML seja a linguagem comercialmente mais popular, existem algumas deficiências de representação nesta linguagem (GUIZZARDI, 2005). A OntoUML é uma linguagem mais acadêmica, pouco conhecida comercialmente, porém foi proposta com o objetivo de suprir as deficiências das linguagens de modelagem conceituais atuais. Apesar de Guizzardi (2005) apresentar em sua tese diversas situações onde a OntoUML é mais expressiva, poucos estudos foram publicados até o momento com relação a avaliação da expressividade da OntoUML em situações práticas.

O objetivo deste experimento foi elaborar um contexto mais próximo a um Sistema de Informação para coletar as percepções entre profissionais e estudantes que não conheciam a linguagem OntoUML. Apesar da falta de conhecimento da linguagem, de uma maneira geral, os participantes indicaram a OntoUML como sendo a linguagem mais expressiva. Além disso, durante a construção dos modelos pelos especialistas, o alto nível de formalidade e consistência induziu aos especialistas em OntoUML a construírem um modelo mais completo do que os especialistas em UML. O experimento mostrou que a OntoUML provoca discussões durante a construção, o que resulta em modelos mais próximos da realidade do domínio a ser representado. Este resultado reforçou a motivação desse estudo no uso de modelos conceituais representados em OntoUML como instrumento para apoiar o processo de elicitação de requisitos.

6.4 Considerações sobre o capítulo

Este capítulo apresentou o resultado de uma avaliação da expressividade de modelos construídos em OntoUML e UML. A avaliação apontou situações onde a OntoUML apresenta maior clareza na representação conceitual do que a UML. Este resultado, em conjunto com a revisão sistemática, reforçou a motivação da tese para o uso de modelos conceituais ontologicamente fundamentados como apoio para o processo de elicitación de requisitos de software. No entanto, é importante conceber um método que permita derivar os requisitos funcionais a partir de modelo conceitual em OntoUML. O próximo capítulo descreve o método proposto e alguns resultados parciais.

A Figura 6-6 resume a etapa da pesquisa em questão e o principal resultado.

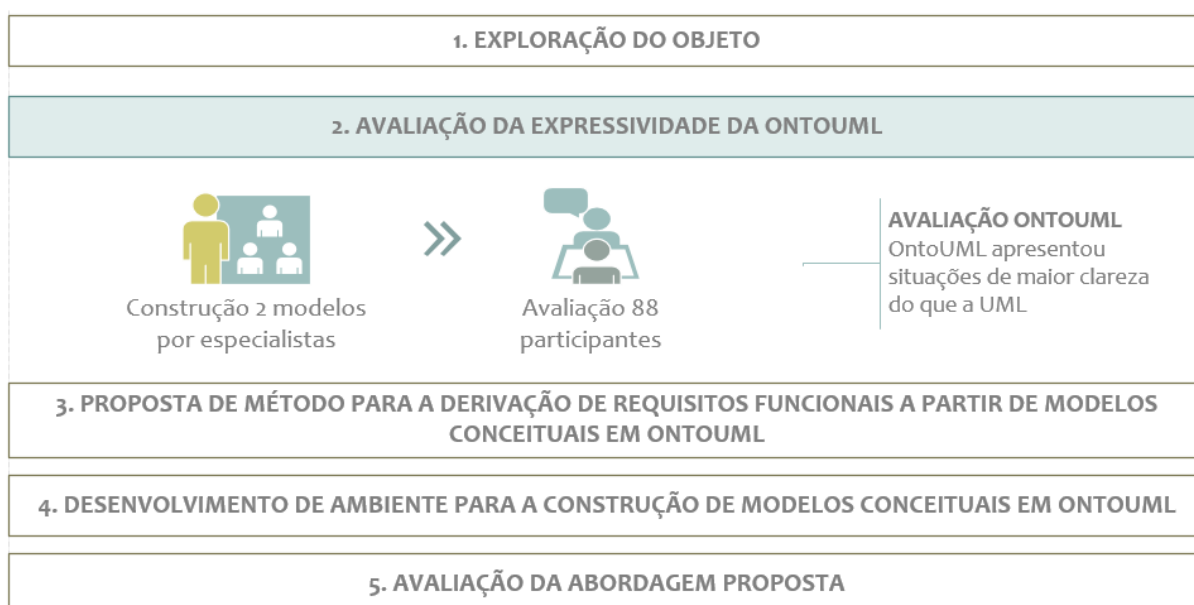


Figura 6-6. 2ª etapa da pesquisa e resultado

CAPÍTULO 7 - MÉTODO PARA A DERIVAÇÃO DE REQUISITOS FUNCIONAIS DE DOMÍNIO

Considerando que os modelos conceituais representados em OntoUML permitem uma representação mais expressiva da realidade, esta tese defende que estes modelos podem ser um instrumento importante para identificar os primeiros requisitos funcionais de um domínio. Neste contexto, o objetivo principal deste capítulo é descrever um método para a derivação de Requisitos Funcionais de Domínio (RFD) a partir de modelos conceituais representados em OntoUML. RFD é uma denominação usada neste estudo para se referir aos requisitos de alto nível gerados a partir da representação de um domínio problema. Este capítulo descreve o método e alguns resultados preliminares. Os resultados desta avaliação também estão publicados em (VALASKI et al., 2017).

A Figura 7-1 situa a etapa da pesquisa correspondente a este capítulo.

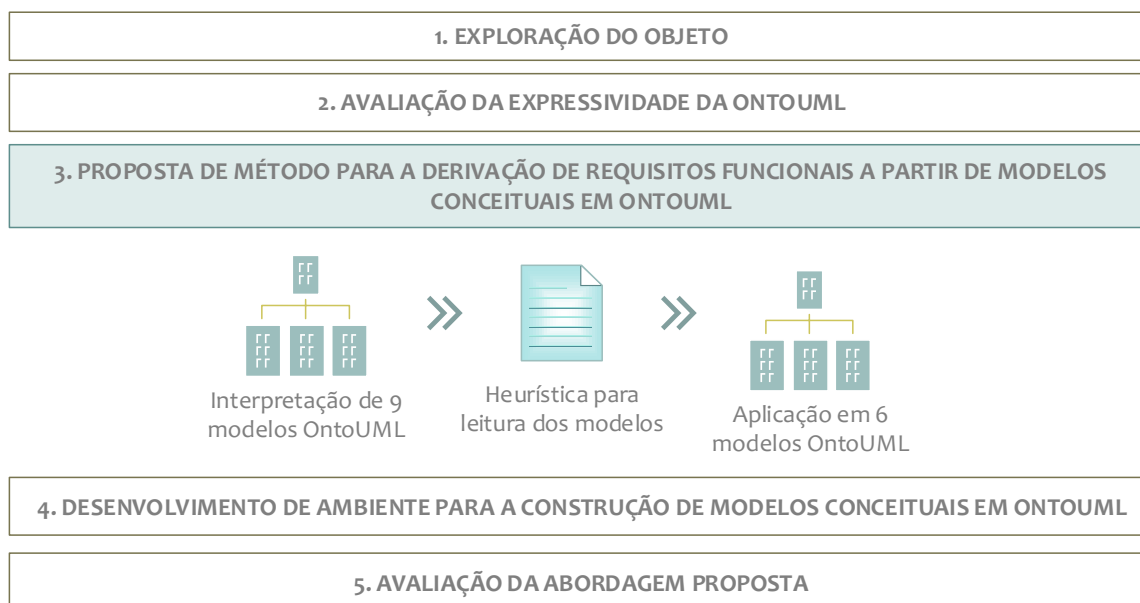


Figura 7-1. 3ª etapa da pesquisa

7.1 Método para a interpretação de um modelo conceitual OntoUML

Um método foi definido com o objetivo de obter uma heurística capaz de derivar sistematicamente os RFD a partir de modelos conceituais em OntoUML. Para obter esta heurística três atividades foram executadas: a seleção de modelos conceituais representados em OntoUML, a transcrição e identificação de padrões para a interpretação dos modelos e a definição da heurística. A seguir detalhes de cada uma destas atividades.

7.1.1 Seleção dos modelos conceituais

O repositório de modelos conceituais em OntoUML denominado Menthor (<http://www.menthor.net/model-repository.html>) foi utilizado como origem para a seleção de alguns modelos conceituais. O Menthor é um repositório criado com o objetivo de centralizar os modelos conceituais em OntoUML publicados na internet por meio de conferências, journal, livros, etc. No total, nove modelos foram selecionados e as principais características de cada modelo são sumarizadas na Tabela 7-1.

Tabela 7-1. Quantidade de elementos do modelo conceitual por domínio

Estereotipo OntoUML	Domínio/Quantidade								
	A	B	C	D	E	F	G	H	I
<i>Category</i>	0	0	0	1	0	0	0	0	0
<i>Collective</i>	0	0	0	7	2	1	1	2	0
<i>Kind</i>	5	4	4	6	2	7	6	6	3
<i>Mixin</i>	0	0	0	0	0	0	1	0	1
<i>Mode</i>	0	0	2	0	0	0	0	0	0
<i>Phase</i>	2	0	2	5	3	2	5	0	2
<i>Relator</i>	2	6	1	1	3	14	3	3	5
<i>Role</i>	6	2	2	2	4	15	9	15	18
<i>Role Mixin</i>	0	0	1	0	0	0	0	0	2
<i>Subkind</i>	0	0	2	2	0	4	17	3	4
<i>Characterization</i>	0	0	2	0	0	0	0	0	0
<i>Formal</i>	0	0	2	4	0	1	1	0	0
<i>Material</i>	3	0	0	6	6	1	1	7	2
<i>Mediation</i>	6	13	2	2	6	37	7	6	15
<i>Generalization</i>	8	2	8	14	6	8	36	18	30
<i>Unknown</i>	0	0	0	0	0	1	2	0	3
Total	32	27	28	50	32	91	89	60	85

Domínios: **A:** Procuração eletrônica; **B:** Roteiro de Ônibus; **C:** Gerenciamento de Projeto; **D:** Instituição de Saúde; **E:** Conferência Acadêmica; **F:** Biblioteca; **G:** Música; **H:** Mentoria Online; **I:** Transporte Escolar

Os estereótipos de classe: *enumeration*, *nominal quality*, *non-perceivable quality*, *perceivable quality* e *quantity* não foram avaliados porque não foram utilizados entre os nove modelos avaliados. Os estereótipos de associação: *componentOf*, *derivation*, *memberOf*, *subCollectionOf* e *subQuantityOf* não foram incluídos pois não apresentaram representação significativa para este contexto, porém, poderão ser considerados em avaliações futuras.

7.1.2 Transcrição e identificação de padrões

Não foi encontrada na literatura um método para a leitura sistematizada de um modelo conceitual, desta maneira um método *ad-hoc* foi proposto, por meio da leitura e transcrição da interpretação dos modelos.

A leitura foi guiada pelos seguintes objetivos: a) identificar regras que permitissem a navegação de todos os elementos do modelo sem repetição, b) transcrever significativamente o conceito e suas relações; c) identificar um padrão que permitisse a transcrição sistematizada. Após diversas leituras e transcrições, alguns padrões foram identificados e são sumarizados a seguir, de acordo com a classe de estereótipos.

- <<Relator>>: é um estereótipo que normalmente agrupa a representação de funcionalidades principais de um domínio. Se a leitura iniciar sempre por um *Relator*, é possível um fluxo de navegação por todos os elementos do modelo. Para a transcrição da funcionalidade relacionada a este estereótipo, foi definida a palavra-chave “gerenciar”;
- <<Category>>, <<Collective>>, <<Kind>>, <<Mixin>>, e <<Subkind>> são estereótipos associados a entidades que requerem a funcionalidade de manutenção de dados. A palavra-chave “manter” foi atribuída para a transcrição da funcionalidade relacionada a estes estereótipos;
- <<Mode>> e <<Phase>> são estereótipos que requerem algum tipo de atualização da sua entidade origem. A palavra-chave “informar” foi definida para a transcrição da funcionalidade relacionada a estes estereótipos;
- <<Role>> e <<Role Mixin>> a princípio, estas classes não necessitam de uma descrição funcional do domínio, porém a relação existente entre este estereótipo e outro elemento, gerará a transcrição de um RFD;
- As relações de associação (<<Characterization>>, <<Formal>>, <<Material>>, <<Mediation>>) geram RFD representando a associação entre

os elementos envolvidos. A palavra-chave “associação” foi atribuída para a transcrição da funcionalidade; e

- As relações de generalização não geram RDF quando a classe envolvida na relação usa os estereótipos: <<Category>>, <<Collective>>, <<Kind>>, <<Mixin>> e <<Subkind>>. Para outras situações, a palavra-chave “associação” também é atribuída para a transcrição da funcionalidade.

É importante ressaltar que a atribuição das palavras-chave associadas aos estereótipos foi realizada de maneira intuitiva com o objetivo de expressar um significado correspondente à uma lista de requisitos de software. Entende-se que a heurística e as palavras-chave podem ser melhoradas conforme o método for aplicado em situações mais amplas.

7.1.3 Definição da heurística

Baseado nos padrões identificados na seção anterior, uma heurística foi definida com o objetivo de gerar sistematicamente a lista de RFD. Na heurística proposta, todas as classes que usam o estereótipo <<Relator>> são selecionadas e armazenadas em um vetor. Para cada uma destas classes, um requisito funcional de domínio é gerado. As classes que usam o estereótipo <<Relator>> levam à identificação de relações entre elementos dependentes. Para cada relação, um requisito funcional de domínio é gerado de acordo com os padrões identificados na Seção 7.1.2.

Para ilustrar a lógica e execução da heurística, um protótipo do algoritmo é apresentado no Quadro 7-1. É importante ressaltar que o algoritmo não representa a versão completa e implementada do programa computacional. Diversas regras são necessárias para o funcionamento completo da heurística proposta. Segue a explicação das funcionalidades principais.

- O algoritmo inicia com o procedimento `principal()` no qual é chamado o procedimento `selecionaTodosRelator()` para a seleção de todas as classes que utilizam o estereótipo <<Relator>>;
- Para cada Relator identificado, será chamado o procedimento `selecionaDependentes()`. Este procedimento filtra os objetos dependentes (nós e relações) de um objeto por meio do procedimento `filtraObjetos()` e imprime os requisitos identificados por meio do procedimento `imprimeRequisitos()`. O procedimento `selecionaDependentes()` é executado

recursivamente até que o nó seja um Relator ou o nó já tenha sido selecionado anteriormente.

Quadro 7-1. Algoritmo parcial para a leitura e extração dos RFD a partir de modelos conceituais em OntoUML.

```

procedimento principal()
inicio
  relator = selecionaTodosRelator();
  para cada relator faça
    escreva("RF...gerenciar" + relator.nome);
    selecionaDependentes(relator);
  fim-para
fim.

procedimento selecionaDependentes(raiz)
inicio
objeto = filtraObjetos(raiz);
  nó = objeto[0];
  relacao = objeto [1];
  para cada nó faça
    imprimeRequisitos(raiz, nó, relacao);
    se (nó.tipo <> 'Relator') ou não(nó.existe)
      selecionaDependentes(nó);
  fim-para
fim.

procedimento imprimeRequisitos(raiz, nó, relacao)
inicio
  se (não(raiz.tipo in ['Kind'] e nó.tipo in ['Phase', 'Subkind']))
  inicio-se
    se (raiz.tipo in ['Relator'] e não(relacao.tipo['Mediation']))
      escreva("RF...gerar" + nó.nome + "de" + raiz.nome);
    senão
      escreva("RF...prover associação de" + nó.nome + "a"
              + raiz.nome);
  fim-se
  senao se (nó.tipo in ['Category','Collective','Kind','SubKind','Mixin'])
    escreva("RF...manter os dados de" + nó.nome);
  senão se (nó.tipo in ['Mode','Phase'])
    escreva("RF...informar" + nó.nome + "a"
            + raiz.nome);
  fim-senão.
fim.
Continued ...

```

- O procedimento `filtraObjetos()` seleciona os objetos (nós e relações) dependentes do objeto raiz. No entanto, para que a navegação do modelo ocorra de maneira unidirecional e apenas as situações que geram RFD sejam selecionadas, o procedimento descarta os objetos nas seguintes situações:
 - Relação igual a *Generalization* e elemento especializado usa estereótipos <<Role>> ou <<RoleMixin>>;

- Relação igual a *Generalization* e o objeto raiz usa o estereótipo <<Subkind>> e o objeto nó usa o estereótipo <<Kind>>;
- Relação igual a *Generalization* e o objeto raiz usa o estereótipo <<Phase>> e o objeto nó usa o estereótipo <<Kind>>;
- Relação igual a *Formal* e o objeto raiz é o destino da relação;
- Relação igual a *Mediation* e o objeto raiz é o destino da relação; e
- Relação igual a *Derivation*;
- O procedimento `imprimeRequisitos()` verifica a natureza do objeto raiz, nó e a relação entre eles para definir como o requisito será impresso.

7.2 Verificação da heurística

A heurística definida na Seção 7.1.3 foi aplicada com seis modelos conceituais entre os nove modelos listados na Tabela 7-1 (A: Procuração Eletrônica; B: Roteiro de Ônibus; C: Gerenciamento de Projeto; D: Instituição de Saúde; E: Conferência Acadêmica; F: Biblioteca). Os modelos selecionados representavam a complexidade (número de elementos e estereótipos) necessária para avaliar a heurística.

7.2.1 RFD derivados a partir da heurística

O Quadro 7-2 e o Quadro 7-3 apresentam a lista parcial de RFD gerada por meio da heurística proposta. O Quadro 7-2 apresenta a lista parcial de RFD para o domínio de Instituição de Saúde (ID: D). A sequência da geração do requisito ajuda a identificar a dependência e a origem entre os requisitos. Os requisitos são listados em cores distintas para enfatizar a origem do estereótipo que gerou o requisito. Em preto os requisitos extraídos de classes que usam o estereótipo <<Relator>>, em azul, os requisitos extraídos de relações (associação e generalização), em roxo os requisitos extraídos e classes que usam os estereótipos: <<Category>>, <<Collective>>, <<Kind>>, <<Mixin>>, e <<Subkind>>, e em laranja os requisitos extraídos de classes que usam os estereótipos: <<Phase>> e <<Mode>>.

Quadro 7-2. RFD parciais relacionados ao domínio de Instituição de Saúde (ID: D).

ID	Descrição do requisito
RF1	O sistema deve gerenciar Consulta Médica.
RF1.1	O sistema deve prover a associação de Profissional de Saúde a Consulta Médica.
RF1.1.1	O sistema deve prover a associação de Pessoa a Profissional da Saúde.
RF1.1.1.1	O sistema deve manter os dados de Pessoa.
RF1.1.2	O sistema deve prover a associação de Unidade de Saúde a Profissional da Saúde.
RF1.1.2.1	O sistema deve manter os dados de Unidade de Saúde.
RF1.1.2.1.1	O sistema deve prover a associação de Região a Unidade de Saúde.
RF1.1.2.1.1.1	O sistema deve manter dados de Região.
RF1.2	O sistema deve prover a associação de Paciente a Consulta Médica.
RF1.2.1	O sistema deve prover a associação de Pessoa a Paciente.
RF2.1	O sistema deve gerar Documento de Consulta Médica.
RF2.1.1	O sistema deve manter os dados de Documento.
continua ...	

A Figura 7-2 ilustra parcialmente o modelo conceitual representado em OntoUML relacionado ao domínio de Instituição de Saúde.

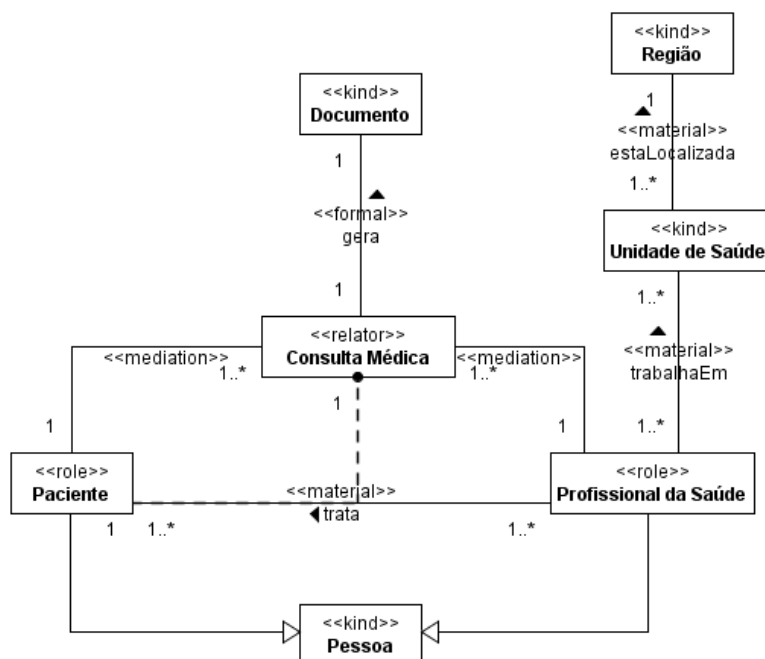


Figura 7-2. Modelo conceitual parcial OntoUML do domínio de Instituição de Saúde (ID: D)

Por meio do Quadro 7-3 é possível observar que os RFDs gerados a partir de classes que usam o estereótipo <<Relator>> são agrupamentos naturais dos demais RFDs. A lista pode sugerir submódulos ou grupos de RFD a partir dos quais use

cases, protótipos de interface, entre outros artefatos de software, podem ser gerados. A Figura 7-3 ilustra parcialmente o modelo conceitual representado em OntoUML relacionado ao domínio de Conferência Acadêmica.

Quadro 7-3. RFD parciais relacionados ao domínio de Conferência Acadêmica (ID: E).

ID	Descrição do requisito
RF1	O sistema deve gerenciar Revisão.
RF1.1	O sistema deve fornecer a associação de Revisor a Revisão.
RF1.1.1	O sistema deve fornecer a associação de Pessoa a Revisor.
RF1.1.1.1	O sistema deve manter os dados de Pessoa.
RF1.2	O sistema deve fornecer a associação de Artigo a Revisão.
RF1.2.1	O sistema deve manter os dados de Artigo.
RF1.2.1.1	O sistema deve informar Artigo não avaliado a Artigo.
RF1.2.1.2	O sistema deve informar Artigo rejeitado a Artigo.
RF1.2.1.3	O sistema deve informar Artigo aceito a artigo.
RF2	O sistema deve gerenciar Submissão.
RF2.1	O sistema deve fornecer a associação de Autor a Submissão.
RF2.1.1	O sistema deve fornecer a associação de Pessoa a Autor.
continua ...	

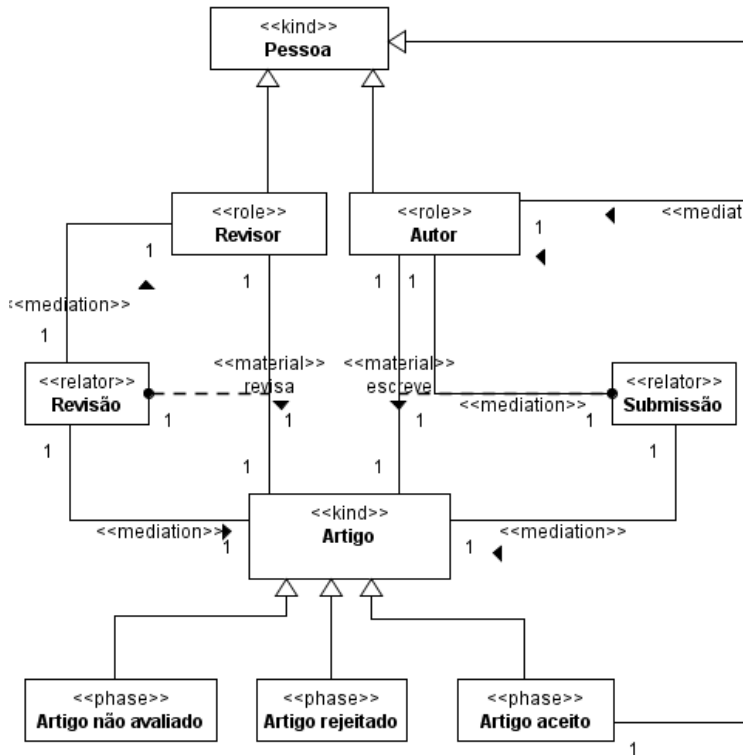


Figura 7-3. Modelo conceitual parcial OntoUML do domínio de Conferência Acadêmica (ID: E).

7.2.2 Verificação dos RFD gerados

Com os resultados da heurística, duas verificações foram realizadas: 1) identificar RFD não listados por falha da heurística; e 2) identificar RFD não listados por falha da modelagem. O objetivo principal destas verificações foi evidenciar a consistência da heurística.

Para a execução destas verificações, três variáveis foram usadas para cada um dos seis domínios: x: número de RFD listados pela heurística; y: número de elementos sem RFD correspondentes; e z: número de elementos do modelo conceitual avaliado. A Tabela 7-2 apresenta os valores para estas variáveis. Primeiramente, foi verificado se RFDs não foram identificados por falha da heurística. Todos os elementos sem RFD listados (y) foram agrupados Tabela 7-3. Após análise individual e manual em cada elemento, chegou-se as seguintes observações:

- A classe que usou o estereótipo <<Collective>> (domínio F) não gerou RFD porque estava associado por meio de uma relação <<MemberOf>>, conforme definido na heurística;
- As classes que usaram os estereótipos <<Role>> e <<Role Mixin>> não geram RFD, conforme definido na heurística;
- As associações que usam o estereótipo <<Material>> relacionados a um <<Relator>> não geram DFR. Nestes casos, a associação que usa o estereótipo <<Mediation>> gerará os RFD necessários, conforme definido na heurística; e
- A relação de generalização não gera RFD, conforme previsto pela heurística.

Tabela 7-2. Variáveis para a verificação da consistência da heurística.

ID	Variável	Domínio/Quantidade					
		A	B	C	D	E	F
x	RFD listados	25	25	22	35	22	71
y	Elementos sem RFD listados	7	2	6	15	10	17
z	Elementos do modelo conceitual	32	27	28	50	32	91

Tabela 7-3. Quantidade de elementos sem RFD listados por domínio.

Estereótipo	Domínio/Quantidade					
	A	B	C	D	E	F
<i>Collective</i>	0	0	0	0	0	1
<i>Role</i>	6	2	2	2	4	15
<i>Role Mixin</i>	0	0	1	0	0	0
<i>Characterization</i>	0	0	0	0	0	0
<i>Material</i>	1	0	0	1	6	1
<i>Generalization</i>	0	0	3	12	0	0
Total	7	2	6	15	10	17

Com estes resultados, foi possível concluir que todas as situações previstas pela heurística ocorreram conforme o esperado. Considerando que a heurística gera no máximo, um requisito funcional de domínio para cada elemento do modelo e a quantidade de elementos sem RFD listados (y) está correta, foi assumido que a soma entre a quantidade de RFD listados (x) e a quantidade de elementos sem RFD (y) deveria ser igual à quantidade de elementos do modelo conceitual (z). De acordo com esta premissa, entre os seis domínios avaliados, apenas o domínio de Biblioteca (ID: F) não é verdadeiro, ou seja, estaria faltando RFDs a serem listados.

Após análise manual, foi possível verificar um requisito funcional de domínio relacionado à uma relação que usou o estereótipo <<Unkown>> não previsto na linguagem OntoUML, ou seja, usado de forma incorreta. Dois RFDs não foram gerados pela presença de uma associação que usou o estereótipo <<mediation>> para relacionar uma classe de estereótipo <<Role>> a uma classe de estereótipo <<Kind>>. Este caso é um problema de *design*, uma vez que a especificação da OntoUML orienta que uma associação que usa o estereótipo <<Mediation>> é previsto ter em sua classe de origem o estereótipo <<Relator>>.

Estes resultados preliminares indicam a possibilidade de usar um modelo conceitual em OntoUML como instrumento de derivação de RFD. Além disso, a heurística pode identificar falhas de design do modelo.

7.3 Considerações sobre o capítulo

Este capítulo apresentou o método proposto para derivar requisitos funcionais de domínio a partir de modelos conceituais em OntoUML. Os resultados preliminares mostraram ser viável a sua utilização. No entanto, novas avaliações em contextos distintos são necessárias para evidenciar a sua utilidade. Além disso, é importante o

desenvolvimento de ferramentas computacionais que apoiem o processo desde a construção do modelo até a derivação dos RFD. O próximo capítulo descreve o ambiente desenvolvido com métodos e as ferramentas integradas para apoiar a abordagem propostas nesta tese.

A Figura 7-4 resume a etapa da pesquisa em questão e o principal resultado.

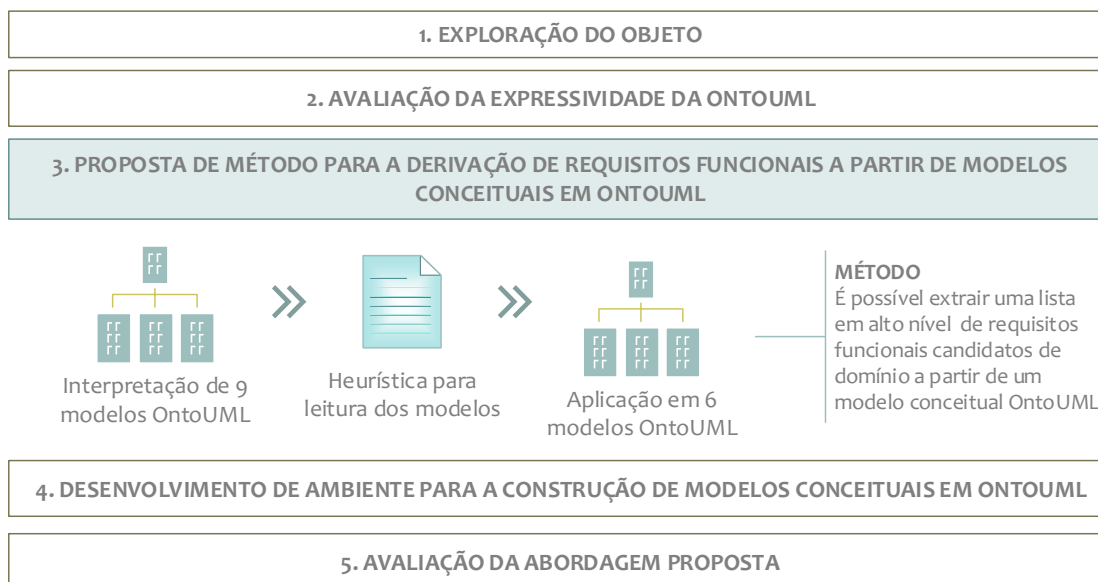


Figura 7-4. 3ª etapa da pesquisa e resultado

CAPÍTULO 8 - AMBIENTE PARA A CONSTRUÇÃO DE MODELOS CONCEITUAIS EM ONTOUML

Conforme apresentado no Capítulo 4, a OntoUML possui um conjunto maior de construtos, os quais possibilitam maior expressividade dos seus modelos conceituais, evitando a sobrecarga e a redundância (GUIZZARDI, 2005). Os resultados apresentados no Capítulo 6 também apontam situações nas quais a OntoUML permite uma representação mais clara do que a UML. No entanto, a utilização da OntoUML é mais complexa do que as demais linguagens, como por exemplo, a UML ou o diagrama ER, principalmente para os modeladores iniciantes (GUIZZARDI et al., 2011).

Uma das dificuldades na construção de um modelo representado em OntoUML é a identificação do construto correto para um determinado conceito a ser representado. Ainda há pouco suporte metodológico para ajudar o usuário a decidir como representar os elementos que denotam propriedades universais em um dado domínio e, muitas vezes, escolhas de modelagem são frequentemente feitas de maneira *ad hoc* (GUIZZARDI, 2005). Guizzardi ressalta a necessidade de métodos que ajudem o usuário a decidir como modelar os elementos de uma conceitualização usando uma linguagem de modelagem na qual os princípios ontológicos estão incorporados na sintaxe.

Este capítulo apresenta o ambiente desenvolvido com o objetivo de apoiar a construção de modelos conceituais usando a linguagem OntoUML. Os resultados deste capítulo também estão publicados em (VALASKI et al., 2014),(VALASKI et al., 2016c). O capítulo inicia com uma breve descrição dos principais trabalhos e conceitos relacionados ao ambiente desenvolvido.

A Figura 8-1 situa a etapa da pesquisa correspondente a este capítulo.

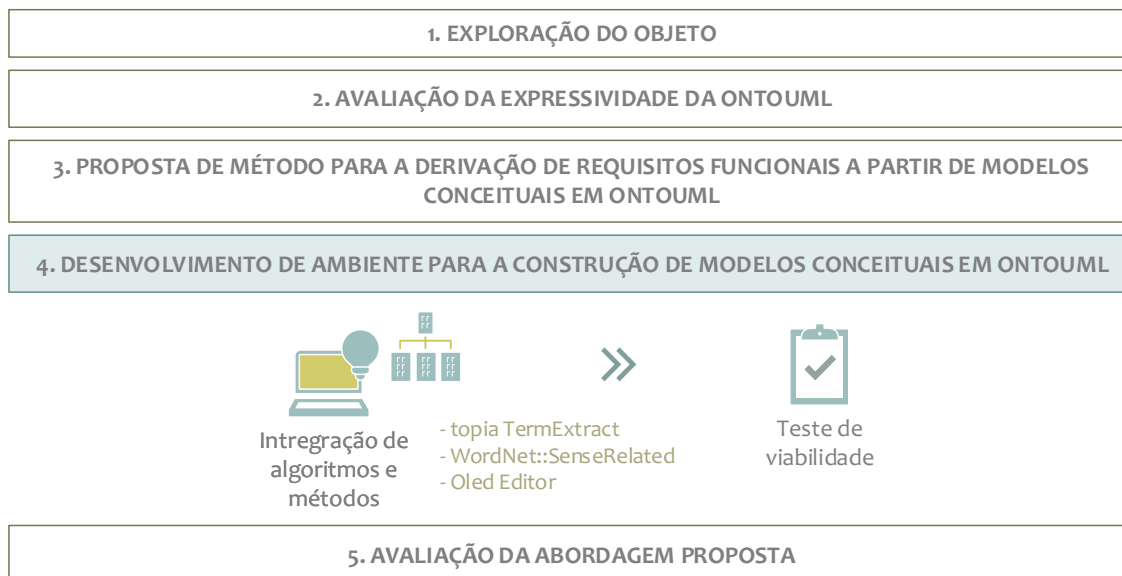


Figura 8-1. 4ª etapa da pesquisa

8.1 Trabalhos e conceitos relacionados

O conhecimento de um universo é sempre transmitido por um discurso e este discurso é enunciado em linguagem natural. A compreensão dos construtos dessa linguagem natural, bem como a semântica, é de vital importância para a criação de modelos mais fidedignos. Castro (CASTRO, 2010) argumenta que os tipos semânticos de Dixon podem ser a base para tal compreensão e propõem uma abordagem linguística para a modelagem conceitual utilizando como linguagem a OntoUML. A abordagem linguística de Castro está baseada no mapeamento entre os tipos semânticos propostos por Dixon e os construtos OntoUML.

8.1.1 Tipos semânticos

Dixon (DIXON, 2005) propõe uma organização semântica para as palavras em classes de significado, denominadas tipos semânticos. Embora a abordagem de Dixon use como exemplo os tipos semânticos do inglês, ele afirma que estes tipos são identificáveis em qualquer língua. Desta maneira, como não há uma proposta semelhante para o português, a classificação de Dixon tem sido utilizada mesmo para a língua portuguesa (CASTRO, 2010).

Os tipos semânticos propostos por Dixon tratam dos substantivos, dos adjetivos e dos verbos. Os substantivos são os tipos semânticos mais significativos

para a modelagem conceitual. Apenas para exemplificar as categorias semânticas, somente os tipos associados aos substantivos serão apresentados neste capítulo.

Para os substantivos são propostas cinco categorias principais: *Abstract Reference*, *Activities*, *Concrete Reference*, *States/Properties* e *Speech Acts*. Dentre estas categorias, a *Concrete Reference* é o mais significativo na modelagem conceitual uma vez que ele nomeia coisas ou seres de existência independente. Cada categoria pode apresentar especializações conforme segue abaixo:

- *Abstract Reference*: ditos de referência abstrata, incluem seres ou coisas de existência dependente e/ou não tangíveis. São especializados em seis subtipos: *Time*, *Place*, *Quantity*, *Variaty*, *Language* e *Other*.
- *Activities*: também de referência abstrata, constituem as atividades. Em sua maioria são derivados de verbos, tais como, venda, compra e corrida.
- *Concrete Reference*: referem-se a seres ou coisas materiais de existência independente. São especializados em 4 subtipos: *Animate*, *Human*, *Parts* e *Inanimate*.
- *States/Properties*: engloba substantivos de referência abstrata cujos referentes são estados físicos ou mentais. Ex: honra, prazer, alegria, dor.
- *Speech Acts*: também de referência abstrata, referem-se a atividades ligadas a fala e tem um verbo associado, como, por exemplo, pergunta e conversa.

Baseado nos tipos semânticos propostos por Dixon, Castro (CASTRO, 2010) propôs um mapeamento dos tipos semânticos para os construtos OntoUML. O Quadro 8-1 apresenta o mapeamento para os substantivos. Em (CASTRO, 2010), é possível consultar o mapeamento completo para os substantivos, os adjetivos e os verbos.

Baseada na proposta de Castro (CASTRO, 2010), Leão et al. (LEÃO et al., 2012) avaliaram ferramentas com a funcionalidade de análise semântica textual. O objetivo da avaliação foi automatizar o processo de identificação dos construtos OntoUML a partir dos tipos semânticos. No entanto, foram identificadas falhas nestas ferramentas. Novos estudos foram realizados por Leão et al. (LEÃO et al., 2013) e concluíram que um dos maiores desafios na identificação automática do tipo semântico é a desambiguação do termo.

Quadro 8-1. Mapeamento dos tipos semânticos associados a substantivos (CASTRO, 2010)

Tipo Semântico		Construto ONTOUML	
<i>Abstract Reference</i>	<i>Time</i>	Datatype	
	<i>Place</i>	Datatype	
	<i>Quantity</i>	Nomes de atributos	
	<i>Variety</i>	Nomes de atributos	
	<i>Language</i>	Análise caso a caso	
	<i>Other</i>	Análise caso a caso	
<i>Activities</i>		Se derivadas de verbos do tipo Social, <i>Contract</i> ou <i>Giving</i> , é um relator	
<i>Concrete Reference</i>	<i>Animate</i>	<i>Kind</i>	
	<i>Human</i>		<i>Kind</i>
		<i>Kin</i>	<i>Role</i>
		<i>Rank</i>	<i>Role</i>
		<i>Social Group</i>	<i>Kind</i>
	<i>Parts</i>	Quando Componente: <i>Kind</i> Quando Ingrediente: <i>Quantity</i> Quando é um Membro: <i>Kind</i> Quando Subcoleção: <i>Collective</i>	
	<i>Inanimate</i>		<i>Kind</i>
		<i>Artefacts</i>	<i>Kind</i>
		<i>Celestial & Weather</i>	<i>Kind</i>
		<i>Environment</i>	<i>Quantity</i>
		<i>Flora</i>	<i>Kind</i>
		<i>Food</i>	Geralmente <i>Quantity</i>
	<i>States</i>		<i>Modes</i>
<i>Speech Acts</i>		Análise caso a caso	

8.1.2 Desambiguação de termos

Uma palavra pode ter vários sentidos e a identificação correta do seu significado pode depender do contexto no qual ela foi empregada. A tarefa de identificar computacionalmente o significado das palavras baseado no contexto que ela ocorre é denominado de *Word Sense Disambiguation* (WSD) (PEDERSEN; KOLHATKAR, 2009). Há técnicas e algoritmos para a desambiguação de um termo, como por exemplo, o Target-Word aplicado para desambiguar apenas uma palavra alvo em uma sentença, e o All-Word aplicado para desambiguar todas as palavras de uma sentença. Um exemplo de uma técnica disponível para a desambiguação é o *WordNet::SenseRelated* (PEDERSEN; KOLHATKAR, 2009). O *WordNet::SenseRelate* é baseado no *WordNet*, que é uma base lexical para a língua inglesa. A ferramenta realiza a desambiguação do termo encontrado na base e

também identifica o tipo semântico correspondente. Para a aplicação da técnica Target-Word, é necessário que sejam identificados os termos alvos. Há técnicas e ferramentas disponíveis para a identificação de termos relevantes em um texto.

Motivados pela possibilidade de desambiguação do termo e a identificação do seu respectivo tipo semântico, Leão et al. (LEÃO et al., 2013) aprimoram o método proposto por Castro (CASTRO, 2010). O novo método se apoia principalmente no uso de uma base semântica para a identificação do tipo semântico.

8.1.3 Bases semânticas

Uma base semântica, além de prover o sinônimo de um determinado termo, também realiza o agrupamento de significado por meio de relações. Uma das principais bases semânticas é a WordNet.Pr (WORDNET.Pr, 2014).

A Wordnet Pr. é uma grande base léxica da língua inglesa desenvolvida pela universidade de Princeton, de onde vem a sigla Pr. Nesta base, os substantivos, os verbos, os adjetivos e os advérbios são agrupados em conjuntos de sinônimos cognitivos (*synsets*), cada um expressando conceitos distintos. Os *synsets* são interligados por meio dos significados (semântica) e relações léxicas. A WordNet é uma base gratuita, disponível para download e com interfaces disponíveis online (WORDNET.Pr, 2014). A base WordNet, a princípio, se assemelha a um tesouro, pois agrupa as palavras baseando-se nos seus significados. Porém, há outras diferenças importantes. A WordNet interliga não somente a forma de uma palavra (conjunto de letras) mas o sentido das palavras. Desta maneira, palavras próximas umas das outras em uma rede são semanticamente desambiguadas.

A maioria das relações na WordNet conectam palavras de mesma classe gramatical. Sendo assim, o WordNet é composto de quatro subredes de substantivos, verbos, adjetivos e advérbios, com alguns ponteiros para classes gramaticais distintas. Além dos termos serem agrupados pelas classes sintáticas (substantivos, verbos, adjetivos e advérbios), eles também são subagrupados pelas classes léxicas, ou também denominados *supersenses*. Para exemplificar, o Quadro 8-2 apresenta o agrupamento para a classe sintática dos substantivos.

Quadro 8-2. Agrupamento léxico do WordNet

Supersenses	Exemplo de conteúdo
<i>Tops</i>	Início dos substantivos
<i>Act</i>	Atos e ações
<i>Animal</i>	Animais
<i>Artifact</i>	Objetos feitos pelo homem
<i>Attribute</i>	Atributos de objetos ou pessoas
<i>Body</i>	Partes do corpo
<i>Cognition</i>	Processos cognitivos
<i>Communication</i>	Processos de comunicação
<i>Event</i>	Eventos naturais
<i>Feeling</i>	Sentimentos e emoções
<i>Food</i>	Comidas e bebidas
<i>Group</i>	Grupos de pessoas ou objetos
<i>Location</i>	Posições espaciais
<i>Motive</i>	Objetivos
<i>Object</i>	Objetos naturais não feitos pelo homem
<i>Person</i>	Pessoas
<i>Phenomenon</i>	Fenômenos naturais
<i>Plant</i>	Plantas
<i>Possession</i>	Posse ou transferência de posse
<i>Process</i>	Processos naturais
<i>Quantity</i>	Quantidade de unidades ou medidas
<i>Relation</i>	Relação entre pessoas ou ideias
<i>Shape</i>	Duas ou três formas dimensionais
<i>State</i>	Estados estáveis
<i>Substance</i>	Substâncias
<i>Time</i>	Tempo e relações temporais

A Figura 8-2 ilustra um exemplo de identificação dos supersenses. Neste exemplo é utilizado o termo “*driver*”, o qual pode ter diferentes significados dependendo do contexto. Para cada significado, um supersense distinto pode estar relacionado, como por exemplo: *person*, *communication* e *artifact*.

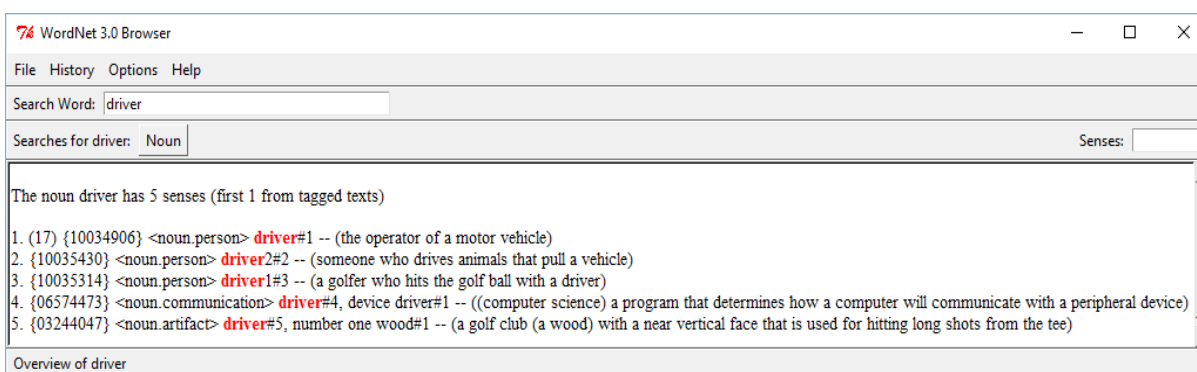


Figura 8-2. Exemplo de supersense identificado pelo WordNet.

Usando o agrupamento léxico do WordNet, Leão (LEÃO, F., 2014) propôs um mapeamento entre os supersenses e os tipos semânticos de Dixon. Para determinadas classes, o mapeamento é direto (Quadro 8-3) e para outras classes, o mapeamento segue regras mais complexas. O mapeamento completo consta na dissertação de Leão (LEÃO, F., 2014).

Quadro 8-3. Mapeamento direto entre os supersenses e os tipos semânticos

Supersense (Wordnet)	Tipo Semântico de Dixon
<i>Act</i>	<i>Activity</i>
<i>Animal</i>	<i>Animate</i>
<i>Body</i>	<i>Part</i>
<i>Cognition</i>	<i>Other</i>
<i>Event</i>	<i>Activity</i>
<i>Location</i>	<i>Place</i>
<i>Motive</i>	<i>Other</i>
<i>Phenomenon</i>	<i>Celestial & Weather</i>
<i>Plant</i>	<i>Flora</i>
<i>Shape</i>	<i>Variety</i>
<i>State</i>	<i>States & Properties</i>
<i>Time</i>	<i>Time</i>
<i>Substance</i>	<i>Environment</i>
<i>Possession</i>	<i>Other</i>
<i>Attribute</i>	<i>States & Properties</i>
<i>Feeling</i>	<i>Other</i>
<i>Process</i>	<i>Activity</i>
<i>Relation</i>	<i>Other</i>

Baseado nos resultados dos trabalhos relacionados à construção de modelos conceituais em OntoUML, um ambiente computacional foi desenvolvido. A seção a seguir descreve os detalhes do ambiente, tais como, as funcionalidades e a arquitetura e posteriormente um teste de viabilidade de uso do ambiente.

8.2 Ambiente computacional desenvolvido

O ambiente foi desenvolvido com o objetivo de apoiar computacionalmente a construção do modelo conceitual em OntoUML e, posteriormente, derivar os requisitos funcionais de um domínio do conhecimento. O ambiente integra algoritmo para a extração de termos relevantes, algoritmo para a desambiguação de termos por meio da WordNet, o editor de OntoUML OLED (OLED, 2014) e um método para a derivação de requisitos funcionais de domínio.

A seguir são apresentadas as funcionalidades que podem ser executadas por meio do ambiente e a arquitetura desenvolvida apresentando os algoritmos, os métodos e as ferramentas integradas.

8.2.1 Funcionalidades do ambiente

Seis atividades principais podem ser executadas por meio do ambiente. As atividades são representadas na Figura 8-3 e descritas a seguir.

- 1) **Identificação de termos relevantes:** a entrada nesta atividade é um texto escrito em linguagem natural. Usando ferramenta de PLN os termos relevantes ser extraídos automaticamente.
- 2) **Criação de arquivo XML:** com os termos relevantes identificados na atividade anterior, um arquivo XML é gerado automaticamente com os termos marcados juntamente com as sentenças nas quais os termos foram aplicados.
- 3) **Desambiguação e extração do *supersense*:** a entrada desta atividade é o arquivo XML com os termos a serem desambiguados. O algoritmo de desambiguação usa a base semântica WordNet para a identificação do sentido do termo de acordo com o contexto de aplicação. Com a identificação do sentido do termo, a base semântica retorna o tipo semântico relacionado ao termo.
- 4) **Mapeamento entre os tipos semânticos e a OntoUML:** com a identificação do tipo semântico correspondente ao termo, é indicado o construto OntoUML baseado no mapeamento proposto por Leão (LEÃO, F., 2014).
- 5) **Construção semiautomática do modelo:** a entrada desta atividade é um conjunto de construtos pré-identificados, a partir dos quais os modeladores complementam o modelo conceitual. Com os conceitos e os respectivos construtos identificados pelo método, é gerado um arquivo .refontoUML. Este arquivo é aberto por meio do editor OLED para o modelador complementar o modelo.
- 6) **Derivação da lista de requisitos funcionais:** a partir do modelo conceitual representado em OntoUML, uma lista inicial dos requisitos funcionais de domínio é gerada.

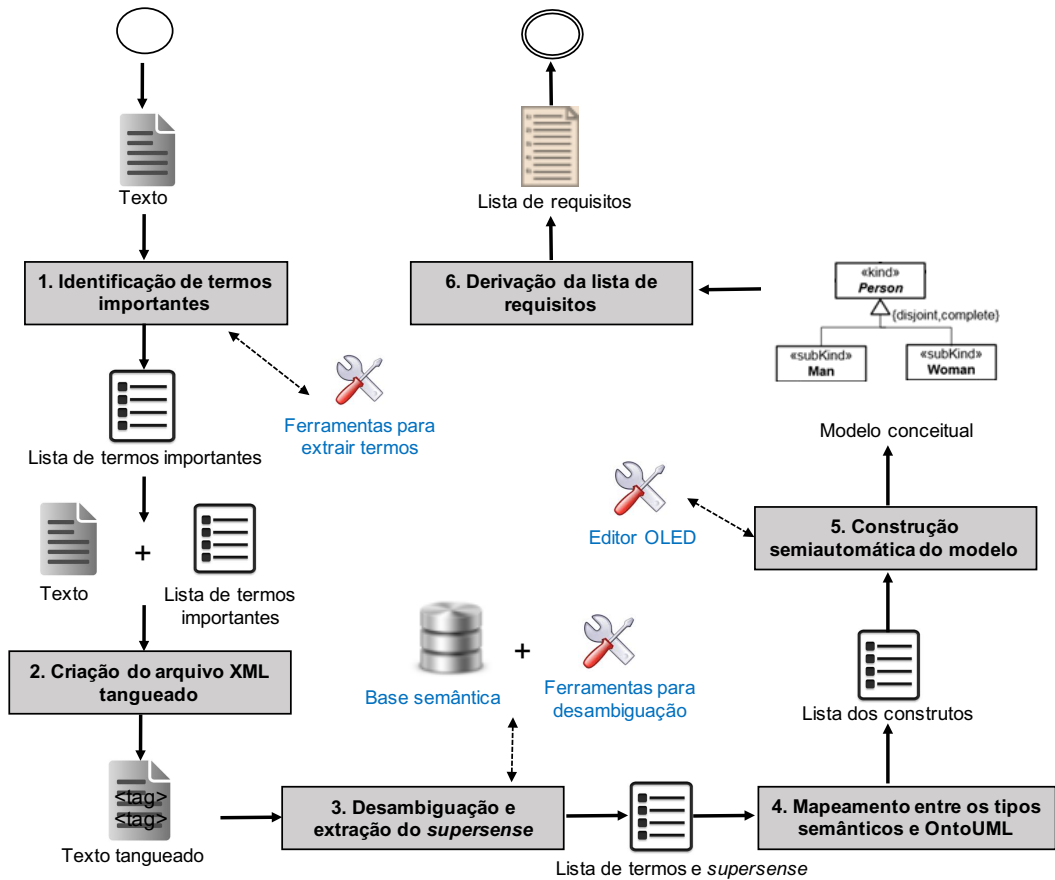


Figura 8-3. Principais atividades executadas por meio do ambiente computacional.

A execução destas atividades é possível devido a um conjunto de algoritmos, ferramentas e métodos integrados ao ambiente. A seguir é apresentada a arquitetura do ambiente.

8.2.2 Arquitetura do ambiente

O ambiente computacional possui uma interface gráfica que possibilita ao usuário interação em todas as atividades descritas na seção anterior. O ambiente foi desenvolvido em linguagem Java e usa para a persistência de dados o banco de dados MySQL. Além disso, o ambiente é integrado por algoritmos, ferramentas e métodos. A seguir são descritas algumas destas integrações.

8.2.2.1 Algoritmo para identificação de termos relevantes

Uma pesquisa exploratória foi realizada (VALASKI et al., 2015) com o objetivo de identificar ferramentas para a extração de termos relevantes. A pesquisa foi realizada usando os mecanismos de busca do Scholar. Por meio de uso de um

conjunto de palavras-chave, a pesquisa inicial retornou 174 publicações e após a aplicação de critérios de inclusão, 25 publicações atenderam o objetivo principal da pesquisa exploratória. Detalhes da pesquisa e resultados podem ser obtidos em (VALASKI et al., 2015).

Nenhuma das 25 publicações selecionadas disponibilizavam uma ferramenta gratuita para utilização. No entanto, os resultados obtidos orientaram sobre as características a serem consideradas ao utilizar estas ferramentas:

- Uso de anotação sintática para extração de substantivos;
- Aplicação da métrica TF-IDF para ranking dos termos relevantes; e
- Identificação de co-ocorrência para ranking de termos compostos.

Por meio de uma pesquisa mais abrangente, utilizando o site de busca Google, foi possível identificar ferramentas disponíveis gratuitamente para a extração de termos relevantes. Estas ferramentas não são citadas nos artigos científicos analisados na pesquisa exploratória. Uma destas ferramentas identificadas foi o `topia.termextract` (TOPIA, 2013).

O `topia.termextract` desenvolvido em Python utiliza abordagem linguística para a identificação de substantivos por meio de anotação sintática. O `topia.termextract` também utiliza abordagem estatística para a identificação de termo composto. Considerando estas características o algoritmo `topia.termextract`, versão 1.1.0, foi selecionado e integrado ao ambiente.

8.2.2.2 Algoritmo para desambiguação dos termos

Para a desambiguação dos termos foi integrado ao ambiente o algoritmo *WordNet::SenseRelated* (PEDERSEN; KOLHATKAR, 2009) desenvolvido em Perl. A técnica utilizada é a *Target-Word* onde é desambiguado apenas a(s) palavra(s) indicada(s) em uma sentença. A entrada do algoritmo é um arquivo XML (Figura 8-4), no qual estão as marcações das sentenças e as palavras alvos. A saída é o *supersense* de cada palavra desambiguada.

```

<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE corpus SYSTEM "lexical-sample.dtd">
<corpus lang='english'>

  <lexelt item="bus.n">
    <instance id="bus.1" docsrc="">
      <context>
        If a <head>bus</head> requires maintenance, Voyager notes the date on which the <head>bus</head> entered m
      </context>
    </instance>
    <instance id="bus.2" docsrc="">
      <context>
        Finally, the average cost to repair all problems with a <head>bus</head> in maintenance is also recorded.
      </context>
    </instance>
  </lexelt>

  <lexelt item="bus drivers.n">
    <instance id="bus drivers.1" docsrc="">
      <context>
        A daily trip is undertaken by one or more <head>bus drivers</head>.
      </context>
    </instance>
  </lexelt>

  <lexelt item="bus trips.n">
    <instance id="bus trips.1" docsrc="">
      <context>
        All <head>bus trips</head> are organized into daily route segments.
      </context>
    </instance>
  </lexelt>

```

Figura 8-4. Arquivo XML para a desambiguação do termo

8.2.2.3 Método para identificação do construto OntoUML

Para a maioria dos casos, o mapeamento Leão (LEÃO, F., 2014) foi usado no ambiente para a identificação do construto OntoUML a partir do *supersense*. No entanto, nem todos os termos são encontrados na WordNet, ou seja, mesmo o termo tendo sido identificado como relevante, não é possível identificar o *supersense* correspondente. Baseado em estudo prévio (VALASKI et al., 2014), foram observadas situações onde seria possível sugerir construtos OntoUML mesmo se o termo não fosse integralmente identificado na base. Para ampliar a possibilidade de sugestão de construtos OntoUML para estes termos, algumas regras foram adicionadas na seguinte ordem:

Regra i: para cada termo identificado, verificar se o termo é composto pelos textos: *number*, *name* ou *date*, se verdadeiro, sugerir o construto datatype;

Regra ii: para os demais termos não atendidos pela regra i, é aplicado o mapeamento de Leão; e

Regra iii: para os demais termos não atendidos pelas regras i e ii, a seguinte regra é aplicada: se termo composto (duas ou mais palavras), identificar se os construtos identificados individualmente são *Kind* ou *Relator*, se verdadeiro, sugerir o construto Relator. Exemplo: **Bus Trips**, este termo composto não existe na base do WordNet. No entanto, os termos **Bus** e **Trip** existem individualmente. Quando o algoritmo Target-Word é executado, é identificado o construto *Kind* para o termo **Bus**

e o construto *Relator* para o termo **Trip**. Neste exemplo, por meio da regra iii é sugerido o construto *Relator* para o termo **Bus Trips**.

8.2.2.4 Ferramenta para a construção do modelo OntoUML

O ambiente gera o modelo conceitual inicial na extensão .refontoUML e faz a sua instanciação por meio do editor de OntoUML OLED. O editor OLED é desenvolvido em Java e sua chamada é feita por meio do .jar correspondente. A partir deste modelo conceitual o modelador complementa conceitos e relações faltantes.

8.2.2.5 Método para a derivação dos requisitos funcionais de domínio

O método proposto no Capítulo 7 foi implementado em Java e integrado ao ambiente. A entrada deste método é um arquivo com a extensão .refontoUML e a saída é uma lista de requisitos funcionais de domínio.

8.3 Teste de viabilidade

Um teste de viabilidade foi conduzido com o objetivo de avaliar o processo de construção semiautomática de um modelo conceitual em OntoUML por meio do ambiente computacional desenvolvido. O teste compreendeu a execução das quatro primeiras atividades propostas na Figura 8-3 (identificação de termos importantes, criação de arquivo XML tagueado, desambiguação e extração do *supersense*, e o mapeamento entre os tipos semânticos e a OntoUML). Todas as atividades foram executadas por meio do ambiente computacional desenvolvido. A seguir são descritas as fases e os resultados do teste de viabilidade. Os resultados também podem ser consultados em Valaski et al. (VALASKI et al., 2016c).

8.3.1 Fases do teste de viabilidade

O teste de viabilidade consistiu basicamente na entrada de uma descrição de domínio para o ambiente computacional e como saída foi obtida uma versão inicial de modelo conceitual em OntoUML. Para avaliar o desempenho dos resultados, foram necessários: uma lista de termos relevantes “ouro” e o modelo conceitual “ouro” correspondentes ao texto selecionado. O texto selecionado para o teste descreve o domínio de roteiro de ônibus em inglês (GEMINO; WAND, 2005) e é apresentado no Quadro 8-4. Por ser um texto publicado cientificamente, reduziu o

viés da elaboração de um texto que pudesse contribuir positivamente ou negativamente com os resultados.

Os termos relevantes, chamados neste teste de termos “ouro”, foram extraídos do modelo conceitual representado em diagrama Entidade/Relacionamento. O modelo conceitual também está publicado no artigo de onde a descrição de domínio foi selecionada (GEMINO; WAND, 2005). Ao total são 32 termos e estão representados no Quadro 8-5.

Quadro 8-4. Texto selecionado (Gemino e Wand, 2005).

Texto
<p><i>There are two ways for people to travel with Voyager. Either passengers can make a reservation on a trip, or passengers can show up at the boarding gate without a reservation and purchase a ticket for an unreserved seat. Passengers with a reservation are assigned a reservation date, whereas, passengers without reservations are assigned a boarding date. The name and addresses of all passengers are collected. Telephone numbers are collected where possible. All bus trips are organized into daily route segments. All daily route segments have both a start time and an end time. Each daily route segment. Voyager organizes is classified as a route segment with a segment number, start town, and finish town. Voyager offers a range of trips, and each trip is made up of one or more route segments. For every trip there is a trip number, start town, and finish town. If the trip is organized around a special event, the event name is also associated with the trip. Each daily route segment that Voyager offers is part of a dally trip. A daily trip is undertaken by one or more bus drivers. The name, address, and employee number of all drivers is collected. Voyager also records information about absent drivers. When a driver is absent. Voyager records the absence start date and the details about the absence. The absent driver provides one or more reasons for being absent and each reason is assigned a detail number and a short description. Voyager also collects information about the buses used for daily trips. Buses have a make, model, and registration number. For buses in use, the average daily kilometers is collected. If a bus requires maintenance, Voyager notes the date on which the bus entered maintenance and records the one or more problems with the bus. Voyager assigns a problem number and a short description for every problems with the bus. Finally, the average cost to repair all problems with a bus in maintenance is also recorded.</i></p>

Quadro 8-5. Termos “ouro” do modelo ER (Gemino and Wand, 2005).

Terms
<p><i>Absence; Absence Start Date; Address; Average Daily Kilometers, Average Cost to Repair; Boarding Date; Bus; Daily Route Segment; Daily Trips; Date Maintenance; Description; Details; Driver; Employee; End time; Finish Town; Maintenance Problems; Make; Model; Name; Passengers; Problem; Registration number; Reservation Date; Route Segment; Segment; Event name; Start Time; Start Town; Telephone; Trip; Trip Number</i></p>

O modelo conceitual “ouro” foi construído manualmente em OntoUML usando a descrição do domínio e está representado na Figura 8-5. Os conceitos relacionados a atributos, no caso da OntoUML, o Datatype, não foram representados por serem considerados menos representativos nesta fase do teste. As métricas *precision* e *recall* (FAWCETT, 2006) foram usadas para avaliar os resultados.

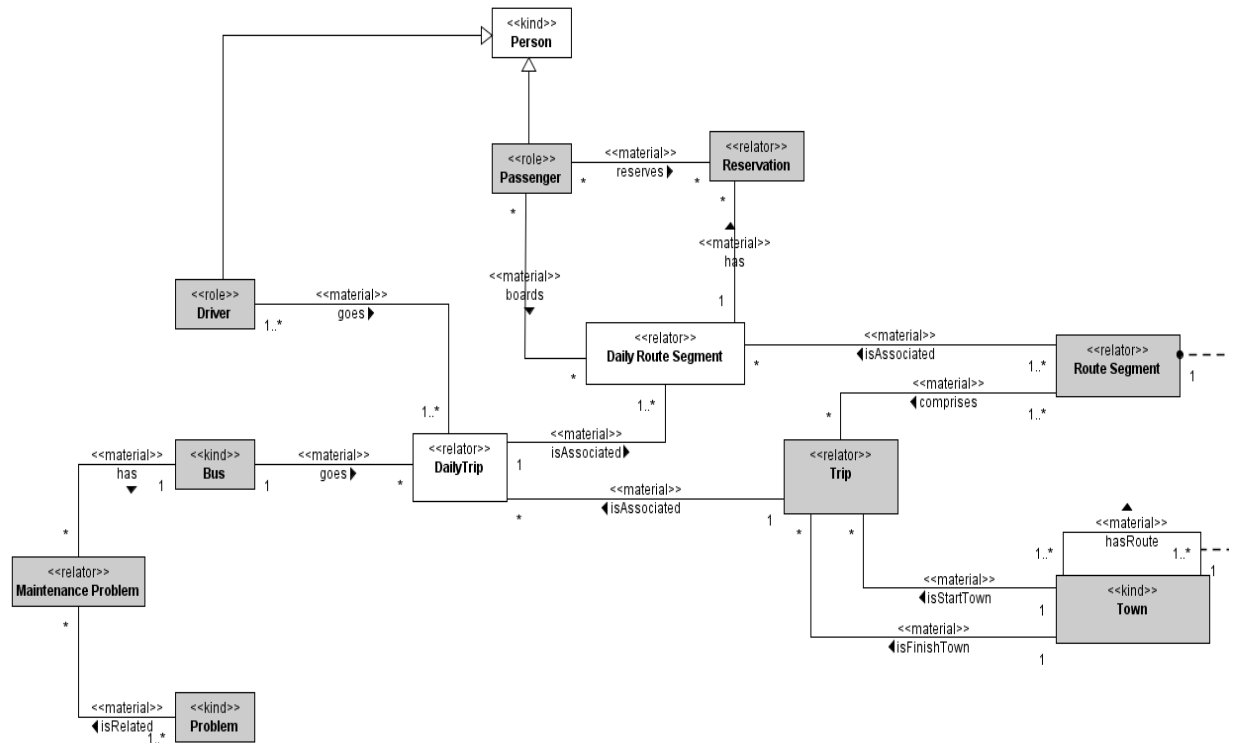


Figura 8-5. Modelo conceitual OntoUML "ouro"

8.3.2 Resultados do teste de viabilidade

Os resultados obtidos por meio da execução do ambiente computacional são apresentados de acordo com os seguintes resultados: identificação dos termos relevantes, identificação dos construtos OntoUML e a construção do modelo conceitual em OntoUML.

8.3.2.1 Identificação dos termos relevantes

A Figura 8-6 e a Figura 8-7 apresentam a sequência de execução para a identificação dos termos relevantes. A Figura 8-6 mostra a seleção do texto previamente cadastrado no ambiente e a Figura 8-7 mostra o resultado da execução do algoritmo topia. O processamento do algoritmo retornou 31 termos relevantes, os quais são apresentados no Quadro 8-6.

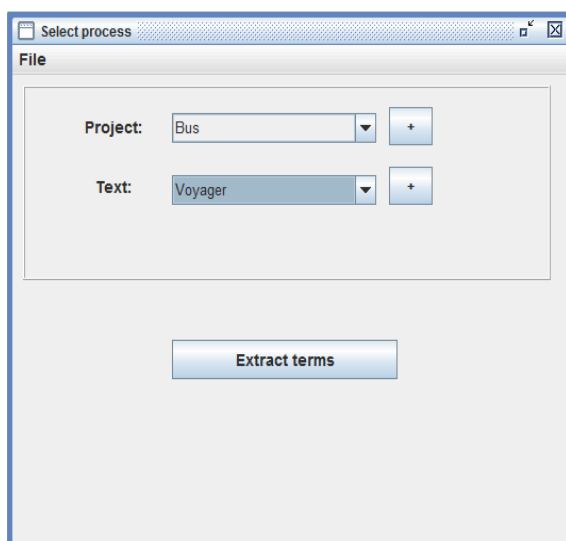


Figura 8-6. Passo 1: seleção do texto

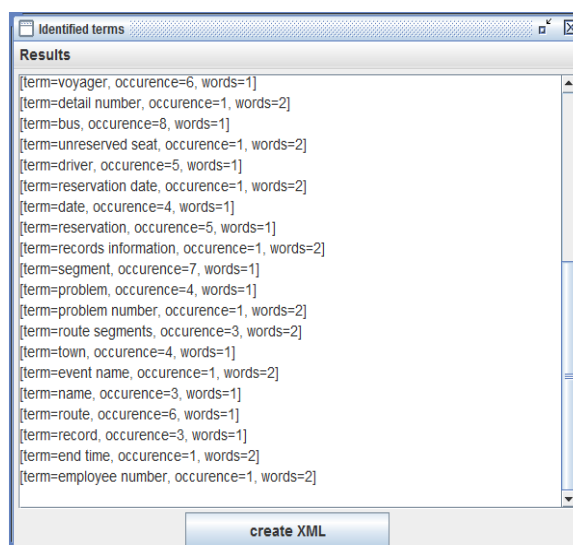


Figura 8-7. Passo 2: resultado do topia

Quadro 8-6. Termos relevantes extraídos pelo Topia versus termos ouro

Termos topia	Termos “ouro” (exato)	Termos “ouro” (parcial)
<i>Bus</i>	<i>Bus</i>	
<i>Bus drivers</i>	-	<i>Bus; driver</i>
<i>Bus trips</i>	-	<i>Bus; trips</i>
<i>Date</i>	-	<i>Boarding date</i>
<i>Detail number</i>	-	<i>Details</i>
<i>Driver</i>	<i>Driver</i>	
<i>Employee number</i>	-	<i>Employee</i>
<i>End time</i>	<i>End time</i>	
<i>Event name</i>	<i>Event name</i>	
<i>Maintenance</i>	-	<i>Maintenance problems</i>
<i>Maintenance problems</i>	<i>Maintenance problems</i>	
<i>Name</i>	<i>Name</i>	
<i>Number</i>	-	<i>Registration number</i>
<i>Passenger</i>	<i>Passenger</i>	
<i>Problem</i>	<i>Problem</i>	
<i>Problem number</i>	-	<i>Problem</i>
<i>Record</i>	-	-
<i>Records information</i>	-	-
<i>Registration number</i>	<i>Registration number</i>	
<i>Reservation</i>	-	<i>Reservation date</i>
<i>Reservation date</i>	<i>Reservation date</i>	
<i>Route</i>	-	<i>Route segment</i>
<i>Route segment</i>	<i>Route segment</i>	
<i>Segment</i>	<i>Segment</i>	
<i>Segment number</i>	-	<i>Segment</i>
<i>Telephone numbers</i>	-	<i>Telephone</i>
<i>Town</i>	-	<i>Finish town; Start town</i>
<i>Trip</i>	<i>Trip</i>	
<i>Trip number</i>	<i>Trip number</i>	
<i>Unreserved seat</i>	-	-
<i>Voyager</i>	-	-

Os resultados apresentados no Quadro 8-6 mostram que entre os 31 termos identificados pelo topia, 14 têm correspondência exata com os termos “ouro”, enquanto que 17 termos não têm correspondência exata. Por outro lado, os resultados apresentados no Quadro 8-7 mostram que entre os 32 termos “ouro”, 14 termos têm correspondência exata com os termos identificados pelo topia, enquanto 18 não têm correspondência exata.

Quadro 8-7. Termos ouro versus termos relevantes topia.

Termos “ouro”	Termos topia (exato)	Termos topia
<i>Absence</i>	-	-
<i>Absence start date</i>	-	-
<i>Address</i>	-	-
<i>Average daily</i>	-	-
<i>Average cost to</i>	-	-
<i>Boarding date</i>	-	<i>Date</i>
<i>Bus</i>	<i>Bus</i>	
<i>Daily route segment</i>	-	<i>Route segment</i>
<i>Daily trips</i>	-	<i>Trip</i>
<i>Date maintenance</i>	-	<i>Maintenance</i>
<i>Description</i>	-	-
<i>Details</i>	-	<i>Detail number</i>
<i>Driver</i>	<i>Driver</i>	
<i>Employee</i>	-	<i>Employee number</i>
<i>End time</i>	<i>End time</i>	
<i>Finish town</i>	-	<i>Town</i>
<i>Maintenance</i>	<i>Maintenance problems</i>	
<i>Make</i>	-	-
<i>Model</i>	-	-
<i>Name</i>	<i>Name</i>	
<i>Passengers</i>	<i>Passenger</i>	
<i>Problem</i>	<i>Problem</i>	
<i>Registration number</i>	<i>Registration number</i>	
<i>Reservation date</i>	<i>Reservation date</i>	
<i>Route segment</i>	<i>Route segment</i>	
<i>Segment</i>	<i>Segment</i>	
<i>Event name</i>	<i>Event name</i>	
<i>Start time</i>	-	-
<i>Start town</i>	-	<i>Town</i>
<i>Telephone</i>	-	<i>Telephone numbers</i>
<i>Trip</i>	<i>Trip</i>	
<i>Trip Number</i>	<i>Trip Number</i>	

Baseado nestes resultados, as métricas *precision* (Fórmula 1) e *recall* (Fórmula 2) foram calculadas. Para o cálculo foram usadas as seguintes variáveis auxiliares: TC (total de termos corretos do topia); nTC (total de termos não corretos do topia); e TGn (total de termos “outro” não retornados pelo topia). Os resultados do cálculo são apresentados a seguir.

$$\text{Precision} = \text{TC}/(\text{TC} + \text{nTC}) = 14/(14 + 17) = 0.4516 \quad (1)$$

$$\text{Recall} = \text{TC}/(\text{TC} + \text{TGn}) = 14/(14 + 18) = 0.4375 \quad (2)$$

A métrica *precision* mostrou que o algoritmo topia obteve uma acurácia de 45.16%, enquanto que a métrica *recall* mostrou que o topia teve uma completude de 43.75%. Estes resultados foram considerados razoáveis, uma vez que somente correspondências exatas foram consideradas no cálculo. Por outro lado, se os termos com correspondência parcial (terceira coluna do Quadro 8-6 e do Quadro 8-7) fossem considerados, os resultados seriam melhores.

Um novo cálculo foi realizado considerando as correspondências parciais. O topia extraiu 27 termos com correspondência parcial entre os termos “ouro”, enquanto que apenas 4 termos (*Record; Records information; Unreserved seat*) não tiveram correspondência parcial. Somente 9 termos (*Absence; Absence start date; Address; Average daily kilometers; Average cost to repair; description; make; model; Start time*) entre os termos “ouro” não tiveram correspondência parcial. Os resultados do novo cálculo (Fórmula 3) (Fórmula 4) são apresentados a seguir.

$$\text{Precision} = \text{TC}/(\text{TC} + \text{nTC}) = 27/(27 + 4) = 0.8709 \quad (3)$$

$$\text{Recall} = \text{TC}/(\text{TC} + \text{TGn}) = 27/(27 + 9) = 0.75 \quad (4)$$

Se os termos parciais fossem considerados para apoiar a identificação dos conceitos relevantes do modelo, o topia teria uma acurácia de 87.09% e completude de 75%. Os resultados foram considerados expressivos para o contexto utilizado neste teste.

8.3.2.2 Identificação dos construtos OntoUML

A Figura 8-8 e a Figura 8-9 apresentam a sequência de execução para a identificação do construto. A Figura 8-8 mostra o arquivo XML criado para realizar a desambiguação. A Figura 8-9 mostra os termos e os construtos OntoUML sugeridos pelo ambiente, após a aplicação das regras i, ii e iii descritas na Seção 8.2.23.

Entre os 31 termos relevantes extraídos pelo topia, o ambiente sugeriu construtos para 28 termos, o que corresponde a 90% de cobertura. Considerando estes 28 termos, 13 termos tiveram o construto Datatype (regra *i*) sugerido, 12 tiveram construtos sugeridos por meio do processo de desambiguação (regra *ii*) e 3 tiveram construtos sugeridos por meio do tipo semântico identificado em termos simples (regra *iii*).

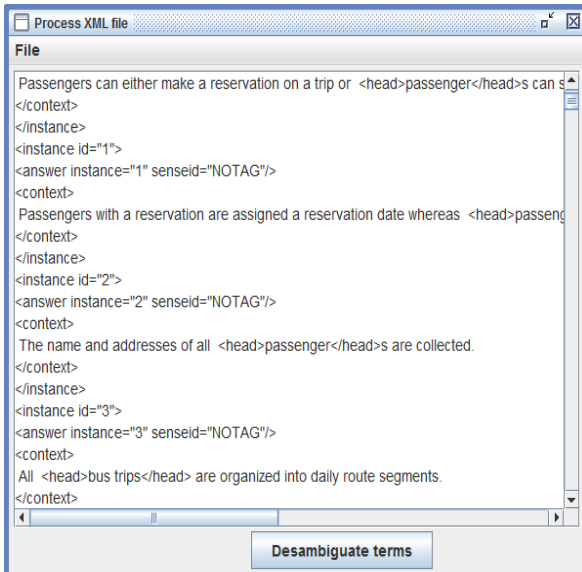


Figura 8-8. Passo 3: arquivo XML criado

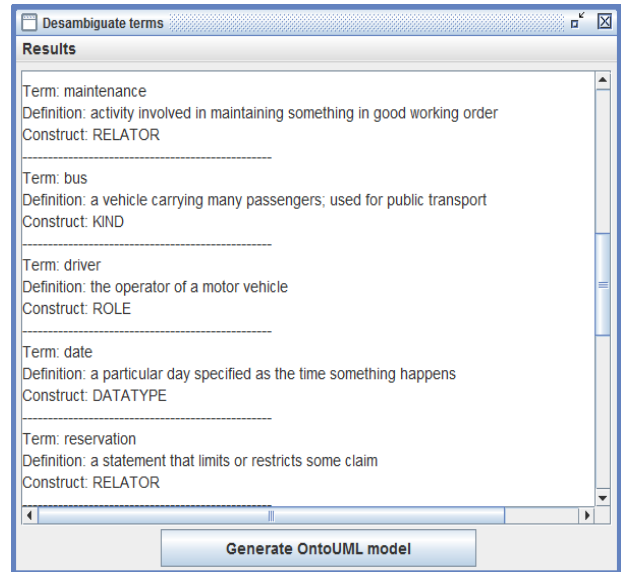


Figura 8-9. Passo 4: desambiguação

8.3.2.3 Construção do modelo conceitual representado em OntoUML

O modelo conceitual foi gerado automaticamente a partir da lista de termos relevantes e seus construtos sugeridos. O modelo conceitual foi instanciado pelo editor OntoUML OLED (Figura 8-10). A identificação automática das relações entre os termos ainda não foi prevista no escopo desta tese.

Considerando apenas os conceitos associados aos construtos *Kind*, *Role* e *Relator*, o ambiente identificou automaticamente 15 conceitos (*Bus*, *Bus drivers*, *Bus Trips*, *Driver*, *Maintenance*, *Maintenance Problems*, *Passengers*, *Problem*, *Reservation*, *Route*, *Route Segment*, *Segment*, *Town*, *Trip*, *Voyager*). Entre estes, 9 estão presentes no modelo conceitual “ouro” Figura 8-5. Estes conceitos estão destacados em cinza. Apenas os termos *Bus drivers*, *Bus Trips*, *Maintenance*, *Route*, *Segment* e *Voyager* não tiveram correspondências no modelo conceitual “ouro”. No entanto, todos estes termos, com exceção do termo *Voyager*, têm correspondência parcial no modelo conceitual “ouro”.

Entre os conceitos representados no modelo conceitual “ouro”, apenas 3 conceitos não foram identificados pelo ambiente computacional (*Person*, *Daily Trip*, e *Daily Route Segment*). No entanto, os conceitos parciais Trip e Route Segment foram identificados. Considerando estes resultados, as métricas *precision* (Formula 5) e *recall* (Formula 6) foram calculadas.

$$Precision = 9/(9+6) = 0.60 \quad (5)$$

$$Recall = 9/(9+3) = 0.75 \quad (6)$$

Neste contexto, considerando apenas os conceitos mais representativos (Kind, Role, Relator) do modelo conceitual “ouro”, o ambiente computacional apresentou uma acurácia de 60% e completude de 75%. Obviamente a qualidade e a complexidade do texto usado para o processamento influenciam diretamente nos resultados.

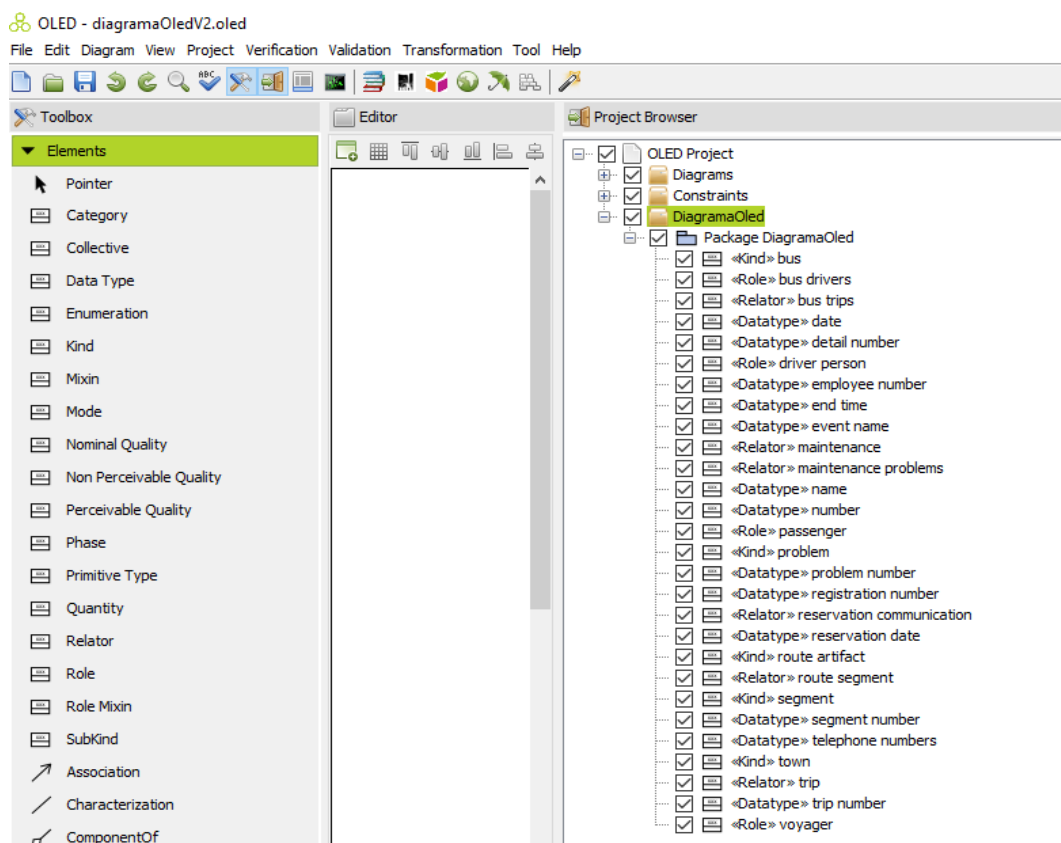


Figura 8-10. Modelo conceitual em OntoUML gerado pelo ambiente computacional

8.3.3 Conclusões do teste de viabilidade

O teste foi executado com o objetivo de avaliar o funcionamento geral do ambiente computacional e, principalmente, identificar os desafios a serem vencidos. De uma maneira geral, o teste mostrou a integração e o funcionamento de diversos algoritmos e métodos que guiam o modelador na construção do modelo conceitual. Na questão operacional, não houve problemas.

Com relação ao desempenho geral do ambiente, no contexto aplicado, o ambiente conseguiu identificar automaticamente 60% dos conceituais significativos de um modelo conceitual. É um resultado que não permite uma afirmação, apenas suposições. Talvez seja um resultado inexpressivo para modeladores experientes, porém seja útil para modeladores em fase de formação, os quais são o usuário-alvo do ambiente desenvolvido.

Com relação ao desempenho de situações específicas, há diversos desafios a serem superados, como, por exemplo, a identificação dos termos relevantes. A precisão da identificação foi baixa, por volta de 45%, quando se consideram apenas correspondências totais. Porém, pode ser melhorada se consideradas as correspondências parciais.

A base WordNet usada para a desambiguação é uma base para conceitos geralmente de domínios genéricos. Ao processar um texto de domínio especializado, com conceitos restritos ao domínio, a desambiguação não será possível e, conseqüentemente, a sugestão do construto OntoUML não será realizada pelo ambiente. Mesmo para os termos que são encontrados e desambiguados, há a possibilidade de a desambiguação não ser feita corretamente. Os algoritmos não têm 100% de precisão.

A atividade de derivação dos requisitos de domínio não foi executada, pois o objetivo principal deste teste foi avaliar a operacionalização das atividades voltadas para a construção semiautomática do modelo conceitual em OntoUML.

8.4 Considerações sobre o Capítulo

Conforme Guizzardi ressaltou, a construção de um modelo conceitual não é uma tarefa trivial, por isso é necessário que se desenvolvam ferramentas de apoio. Este capítulo apresentou um ambiente computacional desenvolvido com o objetivo de dar suporte a construção de um modelo conceitual em OntoUML. O ambiente integra diversos algoritmos e métodos para apoiar esta tarefa. Um teste de viabilidade foi realizado para mostrar a operacionalização e o desempenho do ambiente. O ambiente mostrou-se viável, no entanto, também apontou desafios futuros.

A Figura 8-11 resume a etapa da pesquisa em questão e o principal resultado.

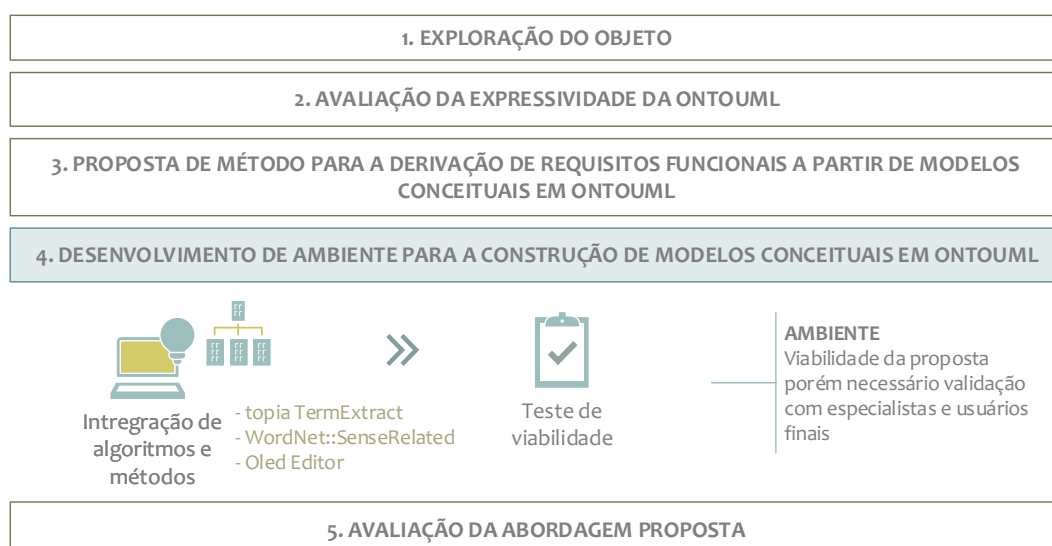


Figura 8-11. 4ª etapa da pesquisa e resultado

CAPÍTULO 9 - AVALIAÇÃO DA ABORDAGEM PROPOSTA

Este capítulo apresenta os resultados relacionados à avaliação da abordagem proposta nesta tese. A avaliação foi conduzida em 2 etapas: a primeira para a identificação automática dos elementos em um modelo conceitual OntoUML e a segunda para a derivação automática de requisitos funcionais de domínio a partir do modelo conceitual OntoUML.

A Figura 9-1 situa a etapa da pesquisa correspondente a este capítulo.

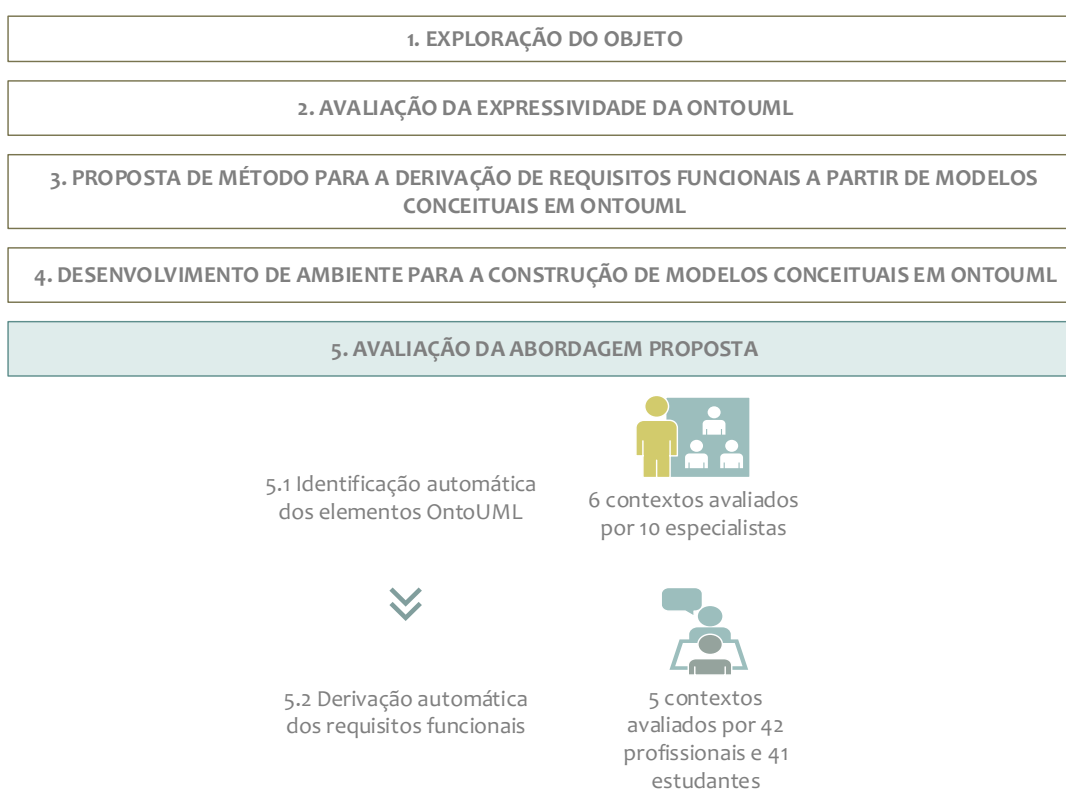


Figura 9-1. 5ª etapa da pesquisa

9.1 Identificação automática dos elementos de modelos conceituais em OntoUML

A primeira etapa da avaliação da abordagem proposta foi conduzida com especialistas em OntoUML para verificar a identificação automática dos elementos em um modelo conceitual OntoUML. Para apoiar a avaliação, 4 métricas foram estabelecidas (Quadro 5-2).

9.1.1 Fases da avaliação

Após a definição do objetivo da avaliação, bem como as métricas para apoiar a avaliação, foram selecionadas as descrições de domínio para a extração de termos relevantes. Para evitar viés relacionado ao tipo de texto aplicado na avaliação, os textos foram selecionados a partir de publicações científicas que discutiam o uso de processamento de linguagem natural com o objetivo de construção de modelo conceitual. Ao total, 6 textos foram selecionados e os dados estão sumarizados na Tabela 9-1. O APÊNDICE D apresenta os textos na íntegra.

Tabela 9-1. Dados sumarizados dos textos utilizados no experimento

ID Texto	Domínio	Qtde Termos	Referência
1.1	Comissão de Vendas	107	(CAPUCHINO et al., 2000)
1.2	Biblioteca	213	(HARMAIN & GAIZAUSKAS, 2003)
1.3	Validação Serviços	243	(DEEPTIMAHANTI & SANYAL, 2011)
2.1	Editora de Livros	76	(MAYR & KOP, 2002)
2.2	Conferência Científica	103	(LEÃO, F., 2014)
2.3	Roteiro de Ônibus	329	(GEMINO; WAND, 2005)

Os textos foram processados seguindo as atividades descritas no Capítulo 8. Algumas alterações de ferramental foram realizadas relacionadas a seleção dos termos relevantes. O algoritmo topia não foi executado neste experimento. Para esta atividade foi utilizada a aplicação LEO (Learning Ontologies System) (LEÃO, F., 2016) desenvolvida no escopo da pesquisa de Leão (LEÃO, F., 2014). O LEO foi utilizado por apresentar em princípio uma abordagem mais eficiente para a identificação dos termos relevantes.

O LEO integra ferramentas de processamento de linguagem natural, tais como: StanforNLP, OpenNLP, LTH Semantic Mapper e WordNet. Este conjunto de ferramentas tem como objetivo reconhecer as unidades básicas de uma sentença. O sujeito, verbo e objeto de cada sentença são considerados os termos relevantes do domínio. Os verbos são identificados para a possibilidade de mapear as relações, porém esta funcionalidade ainda não estava implementada na ferramenta. Ao final do processamento, para cada descrição de domínio, uma lista com os termos relevantes identificados e seu respectivo construto OntoUML foi gerada.

Conforme explicado anteriormente, com estes resultados foram elaborados 2 instrumentos (APÊNDICE B), cada um com o resultado de 3 descrições de domínio distintos. Cada um dos 2 instrumentos foi enviado para 16 especialistas em

OntoUML distintos. A lista inicial dos especialistas foi obtida por meio da seleção de publicações científicas relacionadas a OntoUML. No instrumento, foram apresentadas 2 questões para caracterizar a experiência do participante. Os especialistas, de maneira geral, precisaram indicar se o termo identificado era relevante, se o construto era o mais adequado, e se não fosse, indicar o construto mais adequado. Também foi solicitado aos especialistas identificar os termos relevantes não identificados pelo método.

9.1.2 Resultados da avaliação

A seguir, são apresentados o perfil dos especialistas participantes e o resultado da avaliação.

9.1.2.1 Perfil dos participantes

Entre os 32 especialistas (16 para cada um dos 2 instrumentos), para os quais foram enviados os e-mails, 10 retornaram com as respostas. Cada instrumento, total de 2, foi respondido por 5 especialistas distintos. O perfil dos especialistas em OntoUML está sumarizado na Tabela 9-2. A média geral de experiência dos especialistas foi de 5,3 anos. O nível de conhecimento variou entre intermediário e fluente, com maior número de participantes com nível avançado de conhecimento em OntoUML. Considerando estes dados, pode-se concluir que se tratavam de pessoas com conhecimentos consolidados em OntoUML.

Tabela 9-2. Perfil dos especialistas em OntoUML participantes do experimento

Id instrumento	Id especialista	Qtde anos experiência OntoUML	Nível conhecimento OntoUML
1	1	4	Avançado
1	2	6	Avançado
1	3	7	Avançado
1	4	4	Avançado
1	5	6	Avançado
Média parcial		5,4	
2	1	4	Avançado
2	2	10	Fluente
2	3	7	Fluente
2	4	3	Intermediário
2	5	2	Intermediário
Média parcial		5,2	
Média final		5,3	

A seguir são apresentados os resultados da avaliação.

9.1.2.2 Resultado da avaliação

Os resultados obtidos pelos 10 especialistas foram tabulados e são apresentados em detalhes no APÊNDICE E. Nesta seção são apresentados os resultados sumarizados que apoiaram a avaliação do experimento. A Tabela 9-3 apresenta os resultados consolidados por texto avaliado e por cada medida coletada que apoiaram o cálculo das métricas. Os resultados são apresentados de acordo com o tamanho do texto (do menor para o maior, Tabela 9-1).

Tabela 9-3. Resultado das medidas coletadas para cada texto

Id Texto	Qtde termos					Acerto método (e)
	Relevantes método (a)	Relevantes especialistas (b)	Desambiguados corretamente (c)	Construto OntoUML especialistas (d)		
2.1	12	12	9	12	12	
2.2	14	12	11	12	9	
1.1	12	12	10	11	11	
1.2	20	17	13	14	12	
1.3	25	20	16	14	13	
2.3	38	34	32	29	26	
Total	121	107	91	92	83	

A Tabela 9-4 apresenta os resultados consolidados de acordo com as 4 métricas definidas.

Tabela 9-4. Resultado das métricas para cada texto

Id texto	Percentual termos				Acerto Método (D)
	Relevantes corretos (A)	Desambiguados corretos (B)	Consenso construto OntoUML (C)		
2.1	100%	75%	100%	100%	
2.2	86%	92%	100%	75%	
1.1	100%	83%	92%	100%	
1.2	85%	76%	82%	86%	
1.3	80%	80%	70%	93%	
2.3	89%	94%	85%	90%	
Total	88%	85%	86%	90%	

Considerando o resultado das 4 métricas, é possível observar que o método para a identificação automática é parcial e que há diversos desafios para se atingir a identificação automática total. Porém, o método apresentou ser um instrumento de apoio relevante. A seguir são discutidos os resultados de cada métrica.

Com relação a métrica “termos relevantes corretos”, Tabela 9-4 coluna (A), a média geral de acerto foi de 88% (variação de 80% a 100%). A princípio, o tamanho do texto não interferiu no resultado, pois percentuais maiores foram obtidos tanto nos textos maiores quanto nos menores. Este resultado indica um percentual de cobertura considerável. No instrumento, também foi solicitado que os especialistas indicassem termos relevantes não identificados pelo método. No total geral, apenas 3 termos foram indicados em consenso. No Apêndice E, a Figura 10.3 apresenta em detalhes os termos relevantes indicados pelos especialistas, mas não identificados pelo método.

Com relação a métrica “termos relevantes desambiguados corretos”, Tabela 9-4 coluna (B), a média geral de acerto foi de 85% (variação de 76% a 94%). Esta medida não foi obtida por meio de especialistas porque não era necessário. A avaliação ocorreu manualmente pelo pesquisador comparando o sentido retornado pelo algoritmo TargetWord e o sentido correto no WordNet. Neste ponto do processo, houve a intervenção manual, pois em testes prévios, foram observadas situações em que a desambiguação não acontecia corretamente. Se a identificação é feita de maneira incorreta, o método proposto pode fazer a indicação errada do construto. Não foi possível identificar evidências de que o tamanho do texto influenciou nos resultados, pois percentuais maiores foram obtidos em textos maiores. Talvez a qualidade e a especificidade dos termos contribuam positivamente ou negativamente com os resultados. Os termos desambiguados incorretamente não afetaram os resultados dos especialistas pois a intervenção e a correção manual aconteceu antes do envio do instrumento para os especialistas. Este resultado reforça a necessidade de que ocorra uma análise manual do usuário no método na etapa de desambiguação para posteriormente seguir com as próximas etapas do método.

Com relação a métrica “consenso construto OntoUML”, Tabela 9-4 coluna (C), a média geral do consenso foi 86% (variação de 70% e 100%). Este resultado não avalia diretamente o método proposto, porém evidencia que mesmo em se tratando de especialistas a indicação do construto OntoUML não é uma tarefa trivial. Talvez

em situações com modeladores menos experientes, o índice fosse menor, reforçando a motivação desta tese de que métodos devem ser desenvolvidos para apoiar o modelador a apoiar a identificação dos elementos em um modelo conceitual em OntoUML.

Também foi feita uma investigação para tentar identificar a causa da falta de consenso. Entre os 107 termos identificados como relevantes pelos especialistas, Tabela 9-3 coluna (b), para 15 termos não foi possível obter o consenso. Entre esses termos, foi possível identificar as seguintes características: 9 termos compostos, 3 termos com o *supersense* igual a *cognition* e 3 termos com o *supersense* igual a *activity*. Este resultado pode ser um indicativo de que a identificação de construto OntoUML para termos compostos pode ser mais complexa. Além disso, o tipo do *supersense* a qual o termo pertence, também pode indicar maior complexidade na identificação do construto OntoUML.

Finalmente, com relação a métrica “acerto do método”, Tabela 9-4 coluna (D) a média geral do acerto foi de 90% (variação de 75% a 100%). Certamente este percentual seria inferior se comparado com o número de termos identificados inicialmente, porém, só é possível comparar efetivamente usando os termos que em houve consenso entre os especialistas.

Também foi feita uma investigação para tentar identificar as causas da falha do método. Entre os 92 termos que obtiveram consenso entre os especialistas com relação ao construto OntoUML, para 9 termos o método falhou. Entre esses termos foi possível identificar as seguintes características: 7 termos compostos não foram encontrados na base WordNet, 1 termo com o *supersense* *cognition* e 1 termo não foi possível identificar a causa. É mais comum encontrar termos simples na base do WordNet do que termos compostos. Quando o método proposto não encontra o termo na base do WordNet, ele não consegue fazer a desambiguação e conseqüentemente não consegue processar as etapas seguintes do método. Uma alternativa de trabalhos futuros é fazer o processamento individual das palavras que compõem o termo composto.

9.1.3 Conclusões da avaliação

A identificação de termos relevantes atingiu percentuais relevantes, média de 88%, oscilando entre 80% e 100%. A partir desta identificação, os termos são desambiguados, e neste ponto foi percebido que os algoritmos não têm 100% de

precisão na desambiguação. Nesta etapa, há a necessidade de intervenção, pois a desambiguação incorreta pode resultar na sugestão de um construto totalmente diferente. No entanto, uma das maiores dificuldades do método foi a sugestão de construtos para termos compostos. Foi observado que mesmo os especialistas tiveram certa dificuldade em obter consenso em termos compostos. Esta questão deve ser abordada em trabalhos futuros.

Considerando os resultados obtidos no experimento, fica evidente que o método deve funcionar de forma semiautomática para que o usuário possa interferir nos resultados apresentados pelo método em todas as etapas. Porém, de acordo com os textos avaliados, mesmo considerando todos os desafios do processamento automático, é possível atingir percentuais entre 75% e 100% na sugestão de conceitos em um modelo conceitual representado em OntoUML. É importante ressaltar que o método não identifica relações entre os conceitos. Esta questão também deve ser abordada em trabalhos futuros.

9.2 Derivação automática dos requisitos funcionais de domínio a partir de modelos conceituais em OntoUML

A segunda etapa da avaliação da abordagem proposta foi conduzida com dois grupos distintos, o primeiro grupo formado por profissionais com experiência no levantamento de requisitos e o segundo grupo formado por estudantes com pouca experiência no levantamento de requisitos. O objetivo principal do experimento foi avaliar a derivação automática de requisitos funcionais de domínio a partir de modelo conceitual OntoUML.

9.2.1 Fases da avaliação

A seguir são descritos os detalhes da elaboração do instrumento para a realização da avaliação, bem como os participantes envolvidos.

9.2.1.1 Instrumento

Após a definição do objetivo da avaliação, foram selecionados os textos que seriam utilizados para a construção do modelo conceitual e posteriormente a aplicação do método para a derivação dos requisitos de domínio. Foram utilizados os mesmos textos da Tabela 9-1, com exceção do texto 1.3 (Validação de Serviços),

para o qual houve dificuldades de tradução. Para não comprometer os resultados, optou-se pela sua exclusão.

A partir dos resultados obtidos no experimento descrito na Seção 9.1, foram construídos manualmente os modelos conceituais em OntoUML referentes aos 5 textos selecionados para a avaliação. Com os modelos conceituais construídos, foi executado o método de derivação apresentado no Capítulo 7. Para cada modelo conceitual, uma lista de requisitos de domínio foi gerada. Os textos processados pelo método foram em inglês, porém eles foram posteriormente traduzidos para o português para a apresentação aos participantes da avaliação. As listas de requisitos geradas também foram traduzidas para o português.

Com estes resultados, foram elaborados 3 instrumentos (APÊNDICE C), cada um com o resultado de 2 descrições de domínio distintos e apenas um dos instrumentos com 1 texto de domínio. Cada instrumento foi organizado em 3 etapas principais: a caracterização do participante, a avaliação dos textos e do método e a coleta das percepções do experimento.

Na primeira etapa, foram apresentadas as questões de caracterização dos participantes, tais como: área de formação, grau de informação, nível de experiência com levantamento de requisitos, etc.

A segunda etapa teve o objetivo de qualificar os textos utilizados e a precisão do método. A qualificação do texto foi realizada por meio de 3 afirmações que os participantes tinham que responder, sendo: 1) O texto apresentado descreve claramente o domínio/negócio; 2) O texto apresentado descreve um domínio/negócio de fácil entendimento; e 3) O texto apresentado descreve informações úteis para o Levantamento de Requisitos. Para cada afirmação foi usada a escala de likert com 5 opções de resposta, podendo ser escolhida apenas uma: (1) Discordo totalmente, (2) Discordo, (3) Não concordo nem discordo, (4) Concordo e (5) Concordo totalmente.

A precisão do método foi avaliada sob dois aspectos: o percentual de requisitos funcionais incorretos e o percentual de cobertura do método. A identificação do percentual de requisitos funcionais incorretos ocorreu por meio da indicação pelo participante dos requisitos que no entendimento dele não seriam candidatos em um Sistema de Informação. Já a identificação do percentual de cobertura do método ocorreu por meio da questão: “Você considera que o quadro acima representa quantos por cento do total de requisitos que é possível extrair por

meio da leitura do texto?”. As opções apresentadas ao participante para esta questão foram: (1) 0%, (2) 25%, (3) 50%, (4) 75%, (5) 100%.

A terceira e última etapa do instrumento teve o objetivo de identificar as percepções gerais sobre o experimento, por meio das seguintes afirmações: 1) As listas geradas pelo método representaram um conjunto coerente de Requisitos Funcionais candidatos em um Sistema de Informação; 2) As listas geradas pelo método apresentadas de forma hierárquica podem ser úteis para apoiar a extração dos Requisitos Funcionais candidatos em um Sistema de Informação; e 3) Eu utilizaria um método que gerasse a lista de Requisitos Funcionais para apoiar a identificação dos Requisitos Funcionais do Software. Para cada afirmação foi usada a escala de likert com 5 opções de concordância para a resposta, podendo ser escolhida apenas uma: (1) Discordo totalmente, (2) Discordo, (3) Não concordo nem discordo, (4) Concordo e (5) Concordo totalmente. Também nesta última etapa, foram coletadas informações, em campo aberto, sobre as dificuldades dos participantes com o levantamento de requisitos e pontos positivos e negativos do experimento. O tempo de duração do experimento também foi coletado.

9.2.1.2 Participantes

Como explicado anteriormente, a avaliação ocorreu em dois grupos distintos com o objetivo de identificar se o nível de experiência com levantamento de requisitos poderia influenciar nos resultados.

A formação do primeiro grupo, denominado de profissionais, iniciou com uma lista de participantes conhecidos do grupo de pesquisa em Engenharia de Software, ao qual o pesquisador desta tese está vinculado. Chegou-se a uma lista de 40 profissionais. Os instrumentos 1, 2 e 3 (APÊNDICE C), foram enviados por e-mail alternadamente para cada um dos profissionais da lista. No e-mail, além da participação do experimento, foi solicitado que o instrumento fosse encaminhado para colegas que tivessem experiência nas atividades de Engenharia de Requisitos.

A formação do segundo grupo, denominado de estudantes, deu-se com a seleção de 3 turmas de graduação do curso Bacharelado em Sistemas de Informação. As turmas estavam cursando os períodos 5º, 7º e 8º. Todos os estudantes já haviam sido aprovados em duas disciplinas que abordavam o tema Engenharia de Requisitos. Os instrumentos foram impressos e aplicados em sala de aula. A seguir são descritos em detalhes os resultados desta avaliação.

9.2.2 Resultados da avaliação

A seguir são apresentados os resultados relacionados ao perfil dos participantes, a qualidade dos textos, a precisão do método para identificação dos requisitos e as percepções gerais dos participantes ao final do experimento.

9.2.2.1 Perfil dos participantes

Ao total foram coletadas as respostas de 42 profissionais e de 41 estudantes. Com relação ao perfil dos profissionais, a Figura 9-2 apresenta a área de formação. É possível observar que a maioria dos profissionais tem formação na área da computação, porém em cursos variados, tais como tecnólogos, bacharéis e cientistas. Porém, com maior concentração de formação nos cursos de Ciência da Computação (31%) e Bacharelado em Sistemas de Informação (26%). O nível de formação destes profissionais também é bastante variado, conforme pode ser observado na Figura 9-3, sendo 33% com especialização, 29% com graduação e 26% com mestrado. Estes dados demonstram um grupo de profissionais participantes bem diversificado. Entre o grupo de estudantes, é importante enfatizar que nenhum possui graduação na área.

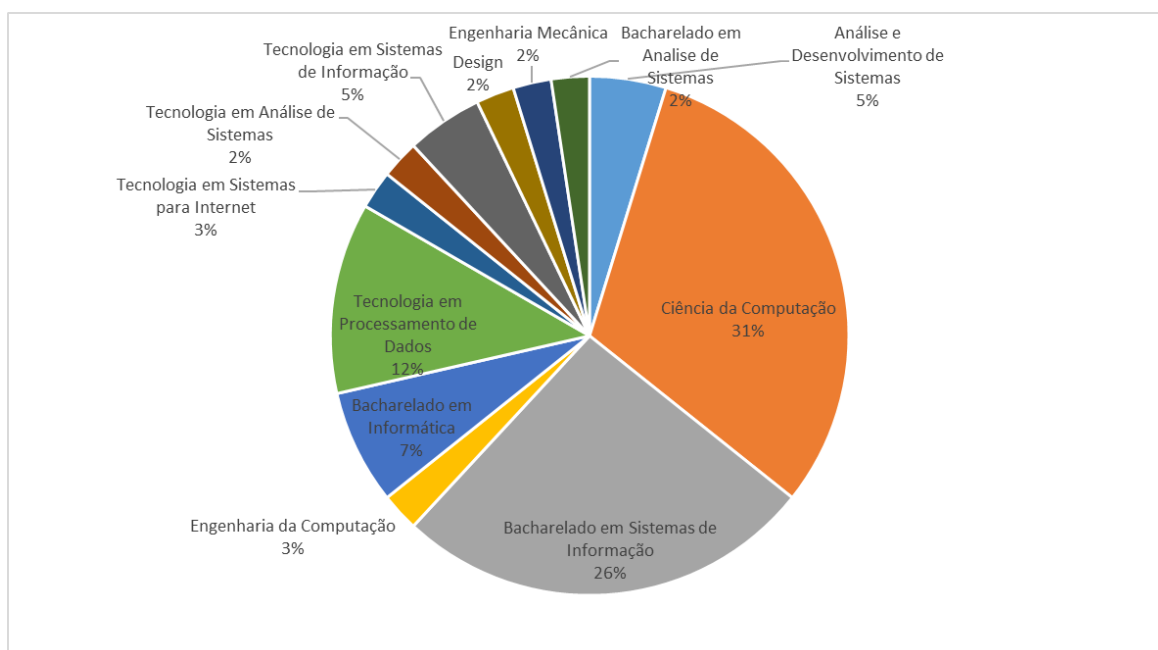


Figura 9-2. Área de formação na graduação dos profissionais

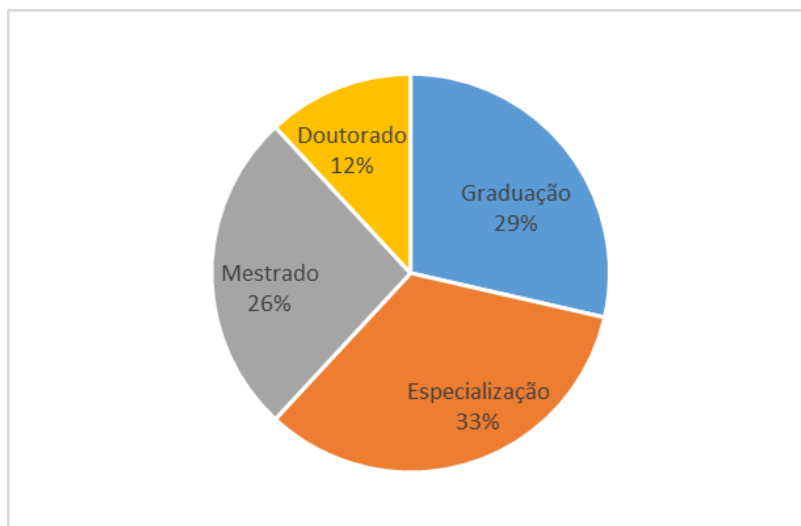


Figura 9-3. Nível de formação acadêmica dos profissionais

Com relação a experiência dos profissionais, entre os 42 participantes, 36 disseram já terem atuado em algum momento da vida profissional recebendo requisitos prontos, ou seja, já levantados para a implementação. Entre estes, a média de anos de experiência foi de 6,69 anos. Já 36 profissionais afirmaram que atuaram em algum momento da vida profissional levantando os requisitos do sistema, sendo a média de anos de experiência de 8,56. A Tabela 9-5 sumariza estas informações. Com relação ao nível de experiência dos profissionais, 24% se consideram muito experientes, 31% se consideram experientes e 31% se consideram razoavelmente experientes. Apenas 14% se consideram pouco experientes. Estes dados estão apresentados na Figura 9-4. Os resultados evidenciam um grupo de participantes com experiência significativa na área de levantamento de requisitos.

Tabela 9-5. Anos de atuação dos profissionais com levantamento de requisitos

Atuação	Qtde participantes	Média de anos
Recebe requisitos prontos	36	6,69
Faz o levantamento de requisitos	36	8,56



Figura 9-4. Nível de experiência dos profissionais com levantamento de requisitos

Com relação a experiência dos estudantes, entre os 41 participantes, 19 disseram já terem atuado em algum momento da vida profissional recebendo requisitos prontos, ou seja, já levantados para a implementação. Entre estes a média de anos de atuação foi de 2,21 anos. Já 15 estudantes disseram já terem atuado em algum momento da vida profissional levantando os requisitos do sistema, sendo a média de anos de experiência de 1,37. A Tabela 9-6 sumariza estas informações. Com relação ao nível de experiência dos estudantes, 68% se disseram pouco experientes, 24% e 7% sem experiência. Ninguém se definiu como experiente ou muito experiente. Estes dados estão apresentados na Figura 9-5. Os resultados confirmam um grupo de participantes com pouca experiência na área de levantamento de requisitos.

Tabela 9-6. Anos de atuação dos estudantes com levantamento de requisitos

Atuação	Qtde participantes	Média de anos
Recebe requisitos prontos	19	2,21
Faz o levantamento de requisitos	15	1,37

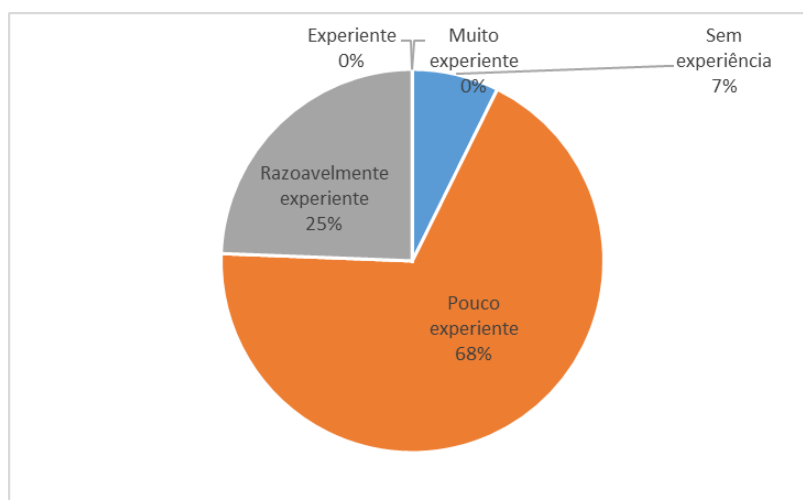


Figura 9-5. Nível de experiência dos estudantes com levantamento de requisitos

9.2.2.2 Qualidade dos textos

Na segunda etapa do instrumento, foi avaliada a qualidade dos textos aplicados na avaliação. A qualidade foi avaliada considerando 3 critérios: a facilidade de entendimento do domínio, a clareza do texto e a utilidade do texto com fins de levantamento de requisitos. Optou-se em fazer esta identificação para avaliar se estes critérios poderiam interferir no resultado do método proposto.

De um modo geral, tanto para os profissionais, Figura 9-6, quanto para os estudantes, Figura 9-7, percentuais elevados de concordância foram obtidos. Em todos os cenários, o percentual atingido considerando as opções concordo e concordo totalmente, foram de pelo menos 68%. O critério que alcançou maior percentual foi com relação a utilidade do texto para fins de levantamento de requisitos. Considerando as opções concordo e concordo totalmente, o resultado foi de 92% para os profissionais e 96% para os estudantes. As únicas diferenças significativas entre os dois grupos foram que para os estudantes, os textos apresentaram mais clareza (82%) do que para os profissionais (68%). Enquanto que os profissionais indicaram maior facilidade do domínio (82%) do que os estudantes (76%). De uma maneira geral, concluiu-se que, os critérios foram atendidos.

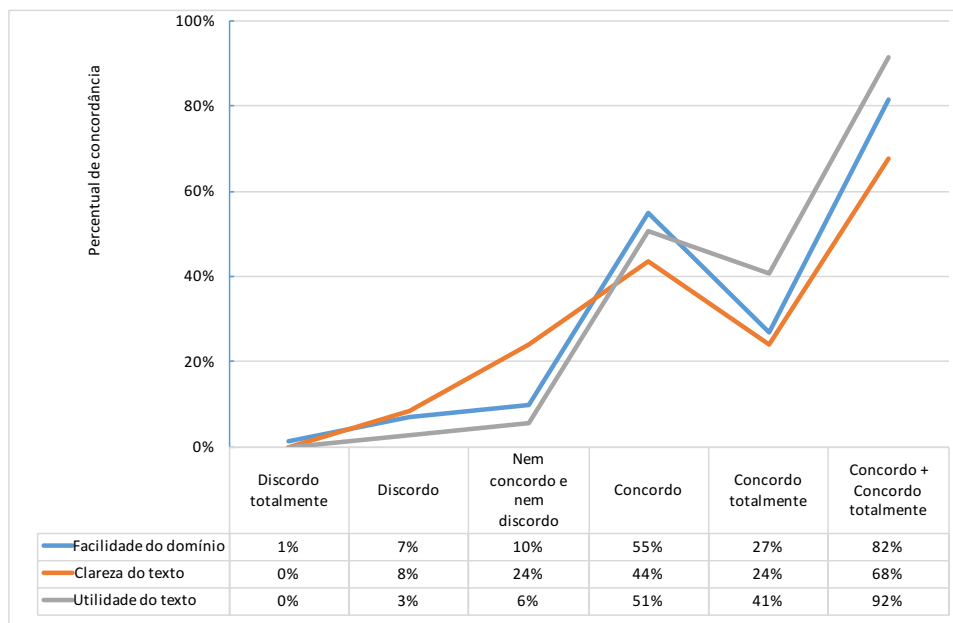


Figura 9-6. Critérios de qualidade dos textos pelos profissionais

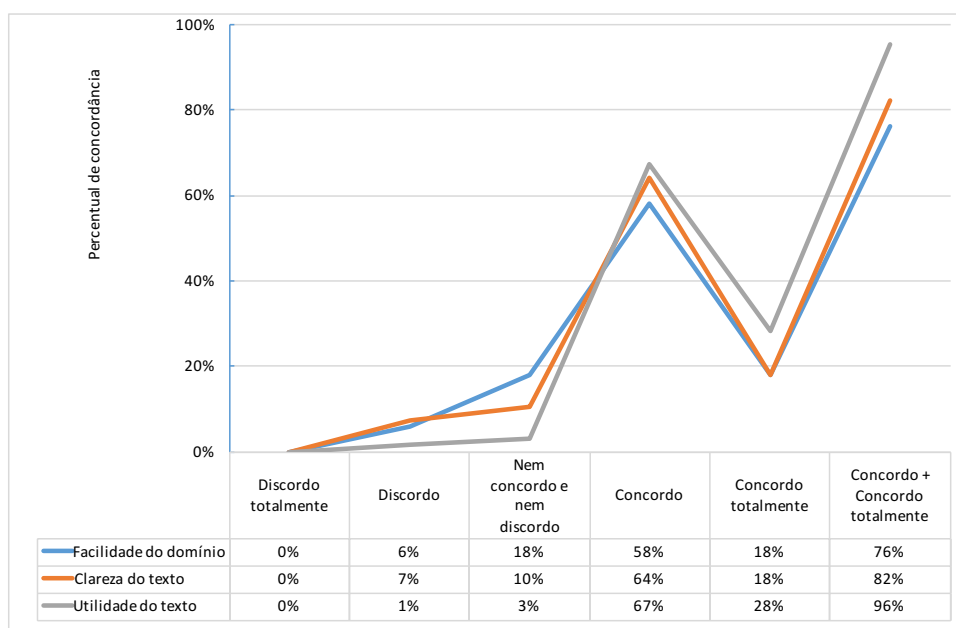


Figura 9-7. Critérios de qualidade dos textos pelos estudantes

Os mesmos critérios foram analisados individualmente para verificar se houve textos onde estes critérios obtiveram percentuais de concordância divergentes. A Tabela 9-7 apresenta em detalhes os resultados dos critérios para cada texto.

Tabela 9-7. Critérios de qualidade por texto e por grupo de participantes

Critério	Grupo	Percentual de concordância	Comissão de vendas	Biblioteca	Editora	Conferência	Roteiro de ônibus
Facilidade do domínio	Profissional	Discordo totalmente	0%	0%	0%	0%	0%
		Discordo	6%	6%	0%	0%	0%
		Nem concordo e nem discordo	6%	6%	0%	8%	8%
		Concordo	56%	44%	54%	46%	54%
		Concordo totalmente	31%	44%	46%	46%	38%
		Concordo + Concordo totalmente	88%	88%	100%	92%	92%
	Estudante	Discordo totalmente	0%	0%	0%	0%	0%
		Discordo	8%	0%	0%	8%	13%
		Nem concordo e nem discordo	23%	23%	23%	15%	7%
		Concordo	62%	54%	54%	69%	53%
		Concordo totalmente	8%	23%	23%	8%	27%
		Concordo + Concordo totalmente	69%	77%	77%	77%	80%
Clareza do texto	Profissional	Discordo totalmente	0%	0%	0%	0%	0%
		Discordo	6%	6%	15%	8%	8%
		Nem concordo e nem discordo	25%	6%	38%	46%	8%
		Concordo	56%	63%	23%	8%	62%
		Concordo totalmente	13%	25%	23%	38%	23%
		Concordo + Concordo totalmente	69%	88%	46%	46%	85%
	Estudante	Discordo totalmente	0%	0%	0%	0%	0%
		Discordo	8%	0%	23%	0%	7%
		Nem concordo e nem discordo	23%	0%	8%	15%	7%
		Concordo	69%	54%	54%	77%	67%
		Concordo totalmente	0%	46%	15%	8%	20%
		Concordo + Concordo totalmente	69%	100%	69%	85%	87%
Utilidade do texto	Profissional	Discordo totalmente	0%	0%	0%	0%	0%
		Discordo	6%	6%	0%	0%	0%
		Nem concordo e nem discordo	6%	6%	0%	8%	8%
		Concordo	56%	44%	54%	46%	54%
		Concordo totalmente	31%	44%	46%	46%	38%
		Concordo + Concordo totalmente	88%	88%	100%	92%	92%
	Estudante	Discordo totalmente	0%	0%	0%	0%	0%
		Discordo	0%	0%	0%	8%	0%
		Nem concordo e nem discordo	8%	0%	8%	0%	0%
		Concordo	69%	54%	69%	69%	73%
		Concordo totalmente	23%	46%	23%	23%	27%
		Concordo + Concordo totalmente	92%	100%	92%	92%	100%

Observando os resultados, para o critério facilidade do domínio, é possível concluir que nenhum domínio foi considerado significativamente difícil. O menor percentual (69%) de concordância foi obtido no texto “Comissão de Vendas” na visão dos estudantes. Já para o critério clareza do texto, os textos “Editora” e “Conferência” tiveram o menor percentual (46%) de concordância na visão dos profissionais.

O texto “Biblioteca” teve o maior percentual (100%) de concordância na visão dos estudantes. Com relação ao critério utilidade do texto, em todos os textos, tanto na visão dos profissionais, quanto dos estudantes, o percentual de concordância foi elevado. O menor percentual (88%) foi obtido pelos textos “Comissão de Vendas” e “Biblioteca” na visão dos profissionais. A seguir são apresentados os resultados referentes a precisão do método para derivação dos requisitos funcionais de domínio.

9.2.2.3 Precisão do método para derivação dos requisitos

Ainda na segunda etapa do instrumento, foi avaliada a precisão do método. Para cada texto apresentado foi avaliada a precisão sob dois aspectos: 1) requisitos funcionais que foram apresentados pelo método, mas que são incorretos e 2) percentual de cobertura do método. Com relação ao primeiro aspecto, o participante indicou de acordo com uma lista de requisitos apresentada pelo método, quais não seriam requisitos candidatos em um Sistema de Informação. A Tabela 9-8 apresenta em detalhes os resultados para cada texto avaliado tanto na visão dos profissionais quanto dos estudantes.

Na Tabela 9-8 são apresentados para cada texto, a quantidade de requisitos gerados pelo método, a quantidade de participantes que avaliaram o texto, a quantidade de escolhas possíveis, considerando as duas informações anteriores, e por fim, a quantidade de requisitos que foram indicados como incorretos pelos participantes. A linha “Parcial de erros” representa o percentual de erros do método para cada texto. No entanto, ao avaliar as situações onde o requisito era identificado como incorreto, na maioria dos casos, era um requisito derivado de uma herança. A Figura 9-8, apresenta um recorte do Instrumento 1 (Apêndice C), referente a lista de requisitos gerada pelo método para o texto “Comissão e Vendas”. Para diversos participantes, os requisitos “Vendedor é um/uma Funcionário”, “Funcionário é um/uma Pessoa”, etc., foram indicados como não sendo requisito

funcional. Na visão dos participantes, a listagem deste requisito não seria necessária.

Tabela 9-8. Percentual de requisitos incorretos gerados pelo método por texto

Grupo	Coleta	Comissão de venda	Biblioteca	Editora	Conferência	Roteiro de ônibus
Profissional	Qtde requisitos gerados pelo método	15	16	10	22	28
	Qtde participantes	16	16	13	13	13
	Qtde escolhas	240	256	130	286	364
	Qtde requisitos indicados como incorreto	45	33	19	40	29
	<i>Parcial de erros</i>	<i>19%</i>	<i>13%</i>	<i>15%</i>	<i>14%</i>	<i>8%</i>
	Qtde requisitos de herança	38	20	17	27	11
	Qtde requisitos indicados como incorreto -					
	Qtde requisitos de herança	7	13	2	13	18
	Final de erros	3%	5%	2%	5%	5%
	Estudante	Qtde requisitos gerados pelo método	15	16	10	22
Qtde participantes		13	13	13	13	15
Qtde escolhas		195	208	130	286	420
Qtde requisitos indicados como incorreto		49	33	37	46	30
<i>Parcial de erros</i>		<i>25%</i>	<i>16%</i>	<i>28%</i>	<i>16%</i>	<i>7%</i>
Qtde requisitos de herança		47	26	33	41	11
Qtde requisitos indicados como incorreto -						
Qtde requisitos de herança		2	7	4	5	19
Final de erros		1%	3%	3%	2%	5%

ID	Descrição do RFD	NÃO é um Requisito Funcional em um SI
RF1	O sistema deve gerenciar Pagamento	
RF1.1	O sistema deve permitir atribuir Vendedor a Pagamento	
RF1.1.1	Vendedor é um/uma Funcionário	
RF1.1.1.1	Funcionário é um/uma Pessoa	
RF1.1.1.1.1	O sistema deve manter os dados de Pessoa	
RF1.1.2	Vendedor é um/uma Empresa	
RF1.1.2.1	O sistema deve manter os dados de Empresa	
RF1.2	O sistema deve permitir atribuir Empresa a Pagamento	
RF2	O sistema deve gerenciar Venda	
RF2.1	O sistema deve permitir atribuir Cliente a Venda	
RF2.1.1	Cliente é um/uma Pessoa	
RF2.2	O sistema deve permitir atribuir Vendedor a Venda	
RF2.3	O sistema deve permitir gerar Ordem de Venda	
RF2.3.1	O sistema deve manter os dados de Ordem de Venda	

Figura 9-8. Recorte Instrumento 1, lista de requisitos gerados (Comissão de Vendas)

Considerando esta observação, os requisitos gerados a partir de uma situação de herança poderiam ser ignorados pelo método, uma vez que não foram considerados relevantes. Para avaliar os resultados com esta alteração, foi contabilizada a quantidade de requisitos de herança, linha “Qtde requisitos de herança”, Tabela 9-8. Com a subtração da quantidade de requisitos de herança da quantidade de requisitos apontados como incorretos, um novo percentual de erros foi recalculado e é apresentado na linha “Final de erros”. Considerando esta nova informação, em nenhum texto, tanto na visão dos profissionais quanto dos estudantes, o percentual de erros foi superior a 5%. Ou seja, os requisitos que são gerados pelo método, em sua maioria são corretos.

O segundo aspecto avaliado foi a cobertura. Embora o método gere uma lista de requisitos com poucos erros, também é necessário verificar qual é a cobertura do método, ou seja, qual é o percentual de requisitos corretos gerados pelo método. Este indicador foi coletado por meio da questão: “Você considera que o quadro acima representa quantos por cento do total de requisitos que é possível extrair por meio da leitura do texto? “. As opções apresentadas ao participante para esta questão foram: (1) 0%, (2) 25%, (3) 50%, (4) 75%, (5) 100%. A Figura 9-9 apresenta o resultado na visão dos profissionais e a Figura 9-10 apresenta o resultado na visão dos estudantes.

Entre os profissionais, 6% consideraram que o método teve uma cobertura de 100% dos requisitos do que é possível identificar por meio da leitura do texto. Enquanto que 58% consideraram que o método tem uma cobertura de 75%, 28% consideraram que o método tem uma cobertura de 50%. Entre os estudantes, 6% consideraram que o método tem uma cobertura de 100% dos requisitos, 52% consideraram que o método tem uma cobertura de 75%, e 22% consideraram que, o método tem uma cobertura de 50%. Com estes resultados, é possível concluir que, minimamente, o método contribui na pior das situações com a identificação de pelo menos 50% dos requisitos funcionais que podem ser identificados por meio da leitura do texto. Também foi possível observar que não houve diferenças significativas entre os dois grupos participantes.

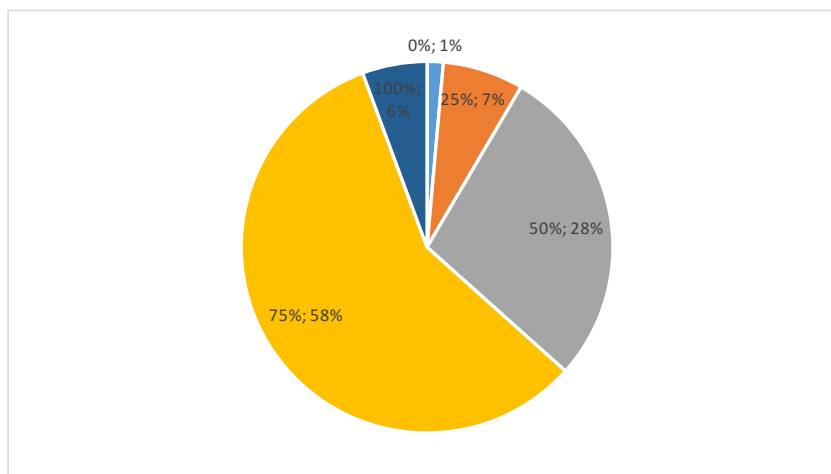


Figura 9-9. Cobertura do método visão geral pelos profissionais

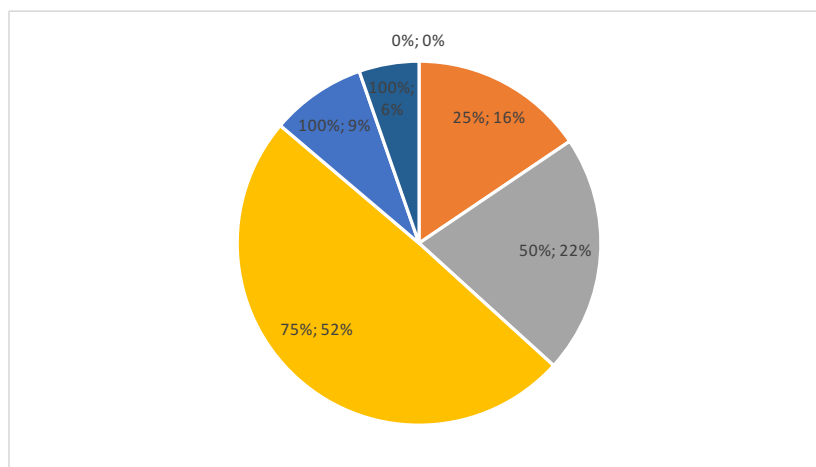


Figura 9-10. Cobertura do método visão geral pelos estudantes

A cobertura do método também foi avaliada individualmente para cada texto com o objetivo de verificar se houve textos com percentuais divergentes. A Tabela 9-9 apresenta estes resultados. Analisando os resultados individualmente, fica evidente que os percentuais obtidos são diferentes para cada texto. Tanto na visão dos profissionais e dos estudantes, os textos “Conferência” e “Roteiro de ônibus” foram os textos em que o método conseguiu maior percentual de cobertura. No texto “Conferência”, 92% dos profissionais e 77% dos estudantes concordam que o método lista pelo menos 75% dos requisitos, enquanto que no texto “Roteiro de ônibus”, 77% dos profissionais e 87% dos estudantes concordam que o método lista pelo menos 75% dos requisitos. São resultados bastante significativos, considerando que o número de requisitos indicados erroneamente foi baixo.

Tabela 9-9. Cobertura do método por texto na visão de profissionais e estudantes

Grupo	Percentual de cobertura	Comissão de vendas	Biblioteca	Editora	Conferência	Roteiro de ônibus
Profissional	0%	6%	0%	0%	0%	0%
	25%	0%	13%	23%	0%	0%
	50%	44%	38%	23%	8%	23%
	75%	50%	50%	54%	69%	69%
	100%	0%	0%	0%	23%	8%
	75% + 100%	50%	50%	54%	92%	77%
Estudante	0%	0%	0%	0%	0%	0%
	25%	31%	31%	8%	8%	7%
	50%	31%	23%	38%	15%	7%
	75%	38%	46%	46%	54%	73%
	100%	0%	0%	8%	23%	13%
	75% + 100%	38%	46%	54%	77%	87%

Embora tenha sido possível identificar diferenças significativas entre os textos, não foi possível confirmar relação entre a qualidade do texto e a cobertura do método. Por exemplo, o texto “ Editora”, no critério “Clareza do Texto”, Tabela 9-7, obteve percentual de concordância de 46% (percentual baixo), segundo os profissionais, e 54% de concordância com relação a cobertura superior a 75%. No entanto, o texto “Biblioteca”, embora tenha obtido o percentual de concordância no critério “Clareza do Texto” (Tabela 9-7) de 88% e 100% (percentual alto) entre os profissionais e estudantes respectivamente, o percentual de cobertura foi menor do que outros textos (50% e 46%). A única relação percebida foi com o tamanho dos textos, textos maiores tiveram maior percentual de cobertura pelo método.

9.2.2.4 Percepção dos participantes sobre o experimento

A terceira e última etapa do instrumento teve o objetivo de identificar as percepções gerais dos participantes com relação a coerência, utilidade e usabilidade do método. Os resultados gerais referentes aos profissionais e aos estudantes são apresentados respectivamente nas Figura 9-11 e Figura 9-12. Os resultados detalhados são discutidos de acordo com as afirmações apresentadas no instrumento.

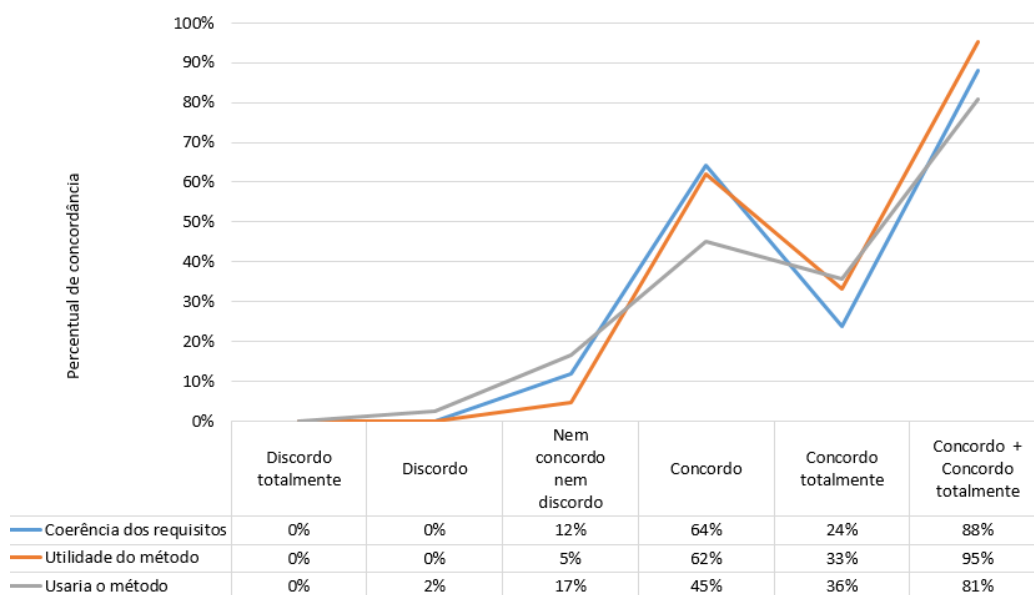


Figura 9-11. Percepções dos profissionais

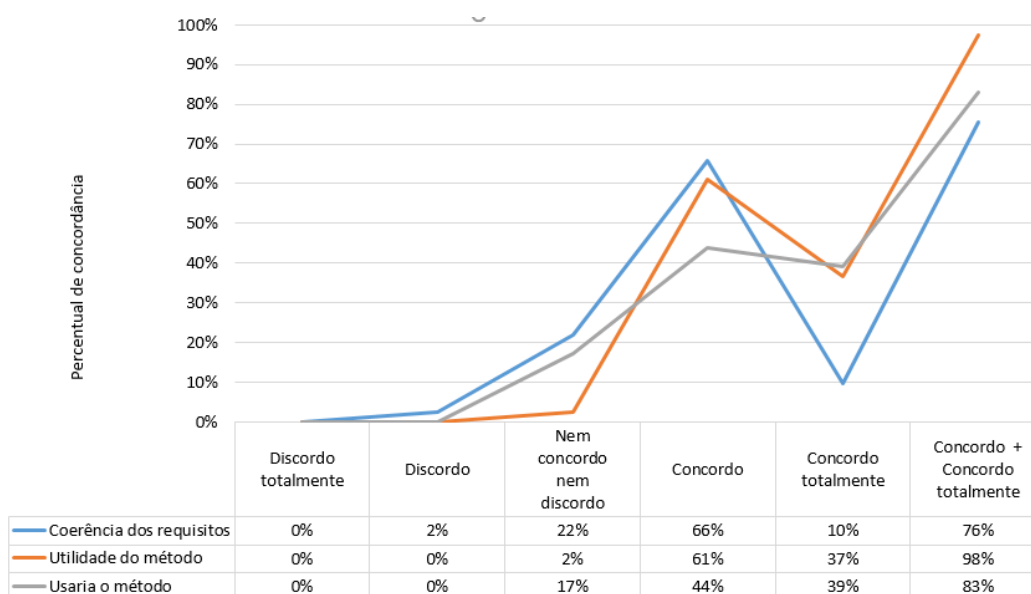


Figura 9-12. Percepções dos estudantes

Afirmção: As listas geradas pelo método representaram um conjunto coerente de Requisitos Funcionais candidatos em um Sistema de Informação.

Segundo os participantes, o método gera uma lista de requisitos coerentes. Acumulando o percentual entre “Concordo” e “Concordo totalmente”, obteve-se um percentual de 88% entre os profissionais e 76% entre os estudantes.

Afirmção: As listas geradas pelo método apresentadas de forma hierárquica podem ser úteis para apoiar a extração dos Requisitos Funcionais candidatos em um Sistema de Informação.

Esta foi a afirmação que atingiu maior percentual de concordância entre os participantes. Acumulando o percentual entre “Concordo” e “Concordo totalmente”, obteve-se um percentual de 95% entre os profissionais e 98% entre os estudantes.

Afirmção: Eu utilizaria um método que gerasse a lista de Requisitos Funcionais para apoiar a identificação dos Requisitos Funcionais do Software.

Acumulando o percentual entre “Concordo” e “Concordo totalmente”, obteve-se um percentual de concordância de 81% para os profissionais e 83% para os estudantes. Embora esta afirmação também tenha atingido percentuais de concordância elevados, eles indicam uma maior tendência em achar o método útil, porém uma menor tendência em usar o método.

Os participantes também puderam preencher, ao final do experimento, observações relacionadas às dificuldades que percebem no levantamento de requisitos e observações relacionadas a realização do experimento. Todas as observações são apresentadas no APÊNDICE F. O tempo médio de execução do experimento para o grupo de profissionais foi de 25 minutos enquanto que no grupo de estudantes foi de 17 minutos. O experimento com os estudantes foi aplicado em sala de aula, enquanto que para os profissionais foi enviado por e-mail. A diferença de tempo pode ser explicada pelo fato dos estudantes estarem em um ambiente controlado, onde não houve interrupção, ou então a maturidade dos profissionais para a resolução com mais cautela.

9.2.3 Conclusões da avaliação

Os resultados obtidos com relação a cobertura do método foram avaliados sob dois aspectos: percentual de requisitos funcionais incorretos e percentual de cobertura do método.

Em todos os 5 textos avaliados, o percentual geral de requisitos funcionais identificados como incorretos foi inferior a 5%. Enquanto que o percentual geral de cobertura, considerando a soma dos resultados das opções 100% e 75% apresentadas aos participantes, ficou em 64% na visão dos profissionais e 61% dos estudantes. No entanto, este mesmo percentual, quando avaliado para cada texto individualmente, obteve uma variação entre 38% (visão dos estudantes no texto “Comissão de Vendas”) e 92% (visão dos profissionais no texto “Conferência”). Ou seja, dependendo do texto, variações significativas foram obtidas. No entanto, não foi possível identificar nenhuma relação direta entre os resultados da cobertura do método e os três critérios de qualidade do texto (facilidade do domínio, clareza do texto e utilidade do texto) definidos na avaliação.

Ao final da avaliação, também foram apresentadas afirmações com o objetivo de avaliar percepções dos participantes. A afirmação *“As listas geradas pelo método representaram um conjunto coerente de Requisitos Funcionais candidatos em um Sistema de Informação”*, obteve um percentual geral de concordância de 88% (profissionais) e 76% (estudantes). Ou seja, a percepção geral dos participantes foi, em sua maioria, de que os resultados do método foram coerentes. A afirmação *“As listas geradas pelo método apresentadas de forma hierárquica podem ser úteis para apoiar a extração dos Requisitos Funcionais candidatos em um Sistema de Informação”*, obteve um percentual geral de concordância de 95% (profissionais) e 98% (estudantes). Ou seja, a percepção geral dos participantes foi, em maioria absoluta, de que o método foi útil. E por fim a afirmação *“Eu utilizaria um método que gerasse a lista de Requisitos Funcionais para apoiar a identificação dos Requisitos Funcionais do Software”*, obteve um percentual geral de 81% (profissionais) e 83% (estudantes). Ou seja, a percepção geral dos participantes foi, em sua maioria, de que utilizaria o método para apoiar a identificação de requisitos funcionais. Os resultados das três afirmações demonstraram que os participantes tiveram uma percepção bastante positiva com relação a coerência, a utilidade e mostraram pré-disposição para usar o método proposto.

É importante ressaltar que o experimento foi conduzido com participantes mais experientes e menos experientes na atividade de levantamento de requisitos. No entanto, os resultados de ambos os grupos de participantes foram muito similares. Todos os resultados, como por exemplo, avaliação da qualidade do texto, cobertura do método e percepções após o experimento, foram muito próximos. A experiência na atividade de levantamento de requisitos não interferiu nos resultados do experimento.

9.3 Considerações sobre o Capítulo

O capítulo apresentou os principais resultados relacionados a avaliação da abordagem proposta nesta tese. A avaliação foi conduzida em 2 etapas. A primeira etapa avaliou a identificação automática de construtos OntoUML e a segunda etapa avaliou a derivação automática dos requisitos funcionais de domínio. Este capítulo encerra a apresentação dos resultados obtidos na condução da pesquisa desta tese. A seguir o fechamento do documento da tese com apresentação das contribuições, limitações e principais conclusões.

A Figura 9-13 resume a etapa da pesquisa em questão e o principal resultado.

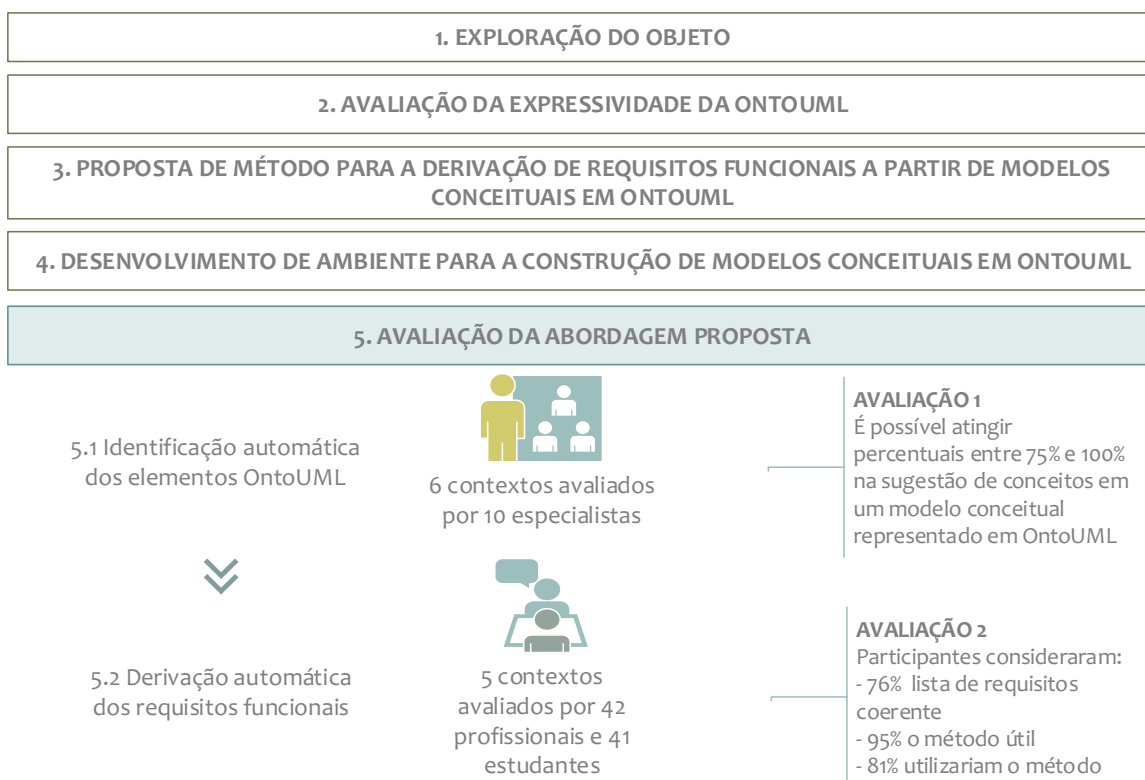


Figura 9-13. 5ª etapa da pesquisa e resultado

CAPÍTULO 10 - CONSIDERAÇÕES FINAIS

Nenhum vento ajuda a quem não sabe para que porto velejar.

Montaigne

Este capítulo tem o objetivo de apresentar as considerações finais referentes aos resultados obtidos nesta tese. A seguir são discutidas a relevância, as contribuições e as limitações da pesquisa. Também é apresentado, ao final do capítulo, possibilidades de trabalhos futuros.

10.1 Relevância da pesquisa

A elicitação de requisitos de software é uma atividade complexa, que demanda qualidade nas comunicações entre os envolvidos para um completo entendimento do domínio do conhecimento a ser elicitado. Requisitos de software mal interpretados podem trazer consequências negativas para todas as demais etapas do processo de desenvolvimento de software. Dificilmente o software entregue ao usuário terá sucesso se as reais necessidades não forem bem compreendidas.

Embora existam diversas técnicas para a elicitação de requisitos, elas ainda são bastante informais e dependem de um esforço humano considerável. A elicitação de requisitos é uma atividade que depende da experiência dos usuários envolvidos. Quando não se pode contar com esta experiência, é comum que ocorram problemas tais como: comunicação deficiente, falta de consenso sobre os termos utilizados e falta do conhecimento do domínio a ser elicitado.

A motivação principal desta pesquisa foi prover apoio à elicitação de requisitos, para auxiliar os usuários, em especial os menos experientes, na captura das reais necessidades dos interessados em relação ao sistema a ser desenvolvido. Para isso, propôs-se o uso de modelo conceitual representado em OntoUML, como instrumento para derivar requisitos funcionais de domínio. O modelo conceitual representado em uma linguagem expressiva possibilita uma representação do domínio mais próxima da realidade. Com isso, informações mais precisas e completas podem ser extraídas deste domínio.

10.2 Contribuições da pesquisa

A Figura 10-1 apresenta uma visão geral das etapas de pesquisa e os principais resultados, os quais são discutidos a seguir.

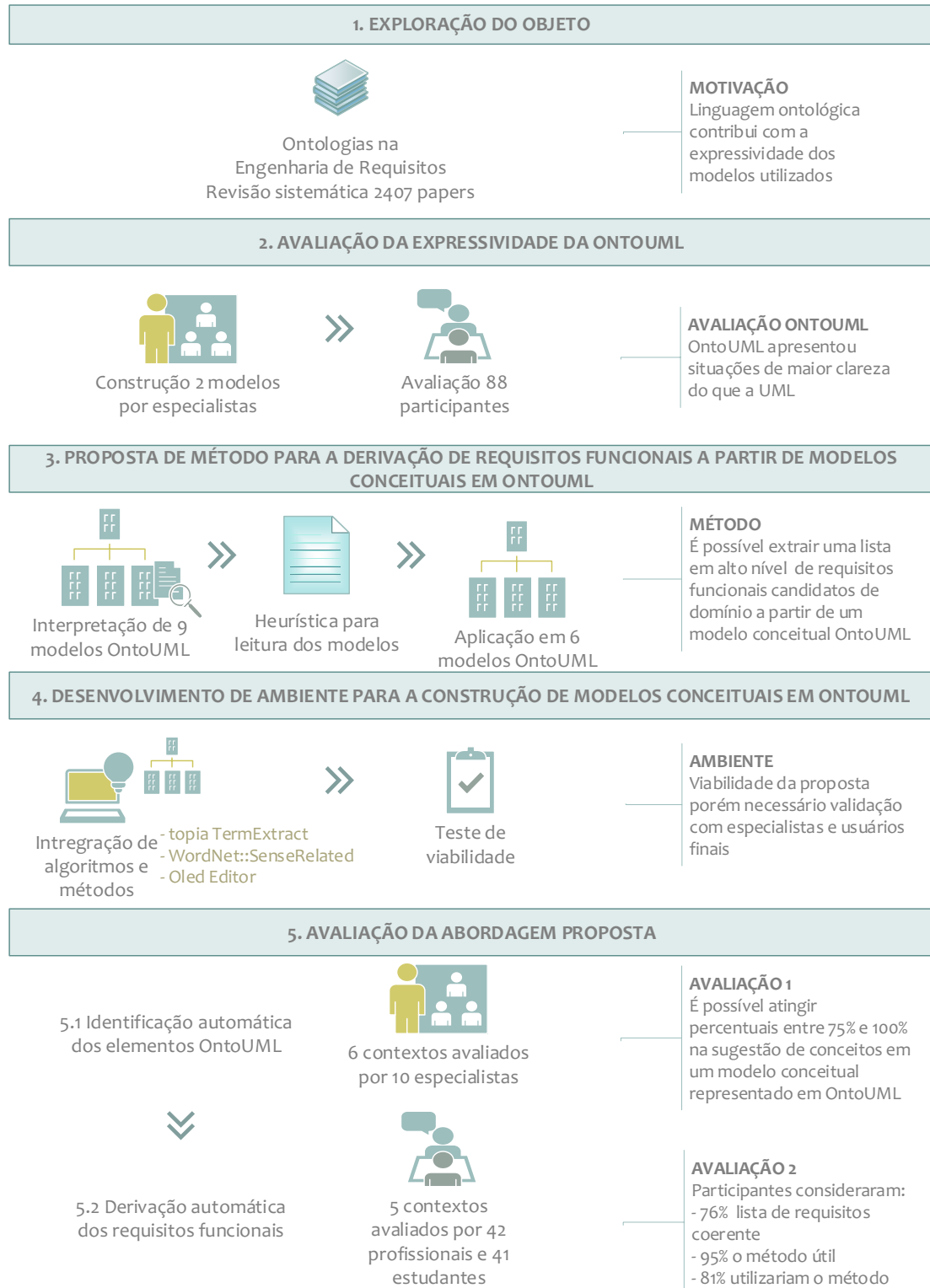


Figura 10-1. Etapas da pesquisa e resultados

Considerando os resultados obtidos por meio da execução desta pesquisa pode-se ressaltar as seguintes contribuições:

- Visão geral, por meio de uma revisão sistemática, de como as ontologias estão sendo aplicadas na área de Engenharia de Requisitos, destacando os objetivos do uso de ontologias não computacionais e do uso de ontologias computacionais. Por meio da revisão sistemática, se evidenciou a importância da utilização de uma linguagem ontologicamente fundamentada para aumentar a expressividade dos modelos conceituais.
- Evidências de situações onde a OntoUML oferece maior clareza do que a UML. Embora Guizzardi tenha demonstrado diversas situações onde a OntoUML permite maior clareza do que as demais linguagens conceituais, não havia sido encontrada uma avaliação em um contexto voltado para Sistemas de Informação e com estudantes que nem conheciam a linguagem OntoUML.
- Método para derivar requisitos funcionais de domínio a partir de modelos conceituais em OntoUML. O método foi automatizado e permite que a partir de um modelo OntoUML, uma lista hierárquica dos requisitos funcionais de domínio de alto nível seja extraída.
- Método para apoiar a identificação automática de conceitos relevantes para um modelo conceitual. Além da identificação dos conceitos, o método sugere o construto OntoUML mais adequado baseado no tipo semântico, identificado por meio de algoritmos de desambiguação.
- Ambiente computacional que apoia a execução das atividades envolvidas na abordagem proposta. O ambiente identifica termos relevantes a partir de uma descrição de domínio e usando diversos algoritmos e ferramentas guia o usuário até a derivação dos requisitos funcionais de domínio.
- Evidências dos pontos críticos que envolvem o processamento de linguagem natural com o objetivo de construir automaticamente um modelo conceitual em OntoUML. Embora os resultados tenham sido expressivos, ainda há desafios a serem superados até que um modelo completo possa ser construído automaticamente.

10.3 Limitações da pesquisa

O objetivo principal desta pesquisa foi propor uma abordagem que apoiasse o uso de modelos conceituais em OntoUML como instrumento para a derivação de requisitos funcionais de domínio. Para atingir este objetivo, um método para apoiar a construção do modelo conceitual e, posteriormente, um método para apoiar a derivação dos requisitos funcionais a partir destes modelos foram propostos e avaliados. No entanto, se reconhece limitações principalmente nas etapas de avaliação. A seguir, são discutidas as principais limitações desta pesquisa.

- As ferramentas de PLN, disponíveis e que produzem melhores resultados, processam textos representados no idioma inglês. Com isso, os experimentos que envolveram a construção do modelo conceitual e, posteriormente, a derivação dos requisitos tiveram que usar textos representados em inglês.
- A disponibilização pública de descrições de domínio limitou a diversidade de textos utilizados nos experimentos. Para evitar o viés da produção de um texto próprio para aplicar no experimento, foram selecionados textos de publicações científicas. No entanto, os textos encontrados, foram textos simples e curtos. O maior texto utilizado no experimento continha 329 palavras.
- Uma das maiores dificuldades na automatização da construção de um modelo conceitual é a identificação de relações entre os construtos. No escopo desta pesquisa, não foi possível trabalhar nesta questão. Este ponto deve ser objeto de trabalhos futuros.
- A lista de requisitos funcionais do domínio gerada pelo método proposto é apenas uma relação em alto nível dos requisitos. A geração da lista tem o principal objetivo de mapear o escopo e o tamanho do domínio. No entanto, o detalhamento destes requisitos, bem como a identificação de regras de negócio, precisa de interações com o usuário.

10.4 Trabalhos futuros

Considerando os resultados desta pesquisa, foram identificadas algumas possibilidades de desdobramentos em trabalho futuros, tais como:

- Propor método para identificação de construtos OntoUML para termos compostos. O método utilizado na tese utiliza a base semântica WordNet para fazer a sugestão, no entanto, termos compostos são mais difíceis de serem encontrados na base.
- Implementar a identificação de relações entre os construtos OntoUML e, assim, propiciar a construção completa do modelo conceitual.
- Realizar experimentos usando descrições de domínio reais a um Sistema de Informação. Com os resultados, avaliar se o método gera requisitos funcionais de domínio próximos aos que foram identificados no projeto de desenvolvimento de software.
- Introduzir o uso de modelos conceituais em OntoUML no processo de levantamento de requisitos e avaliar os benefícios na identificação do escopo, tamanho e desdobramento em outros artefatos.

10.5 Conclusão

O uso do modelo conceitual em OntoUML foi a maior motivação desta tese. Guizzardi (2005) já havia enfatizado em sua tese diversas situações onde a OntoUML é mais expressiva do que outras linguagens.

No entanto, os resultados desta tese também ajudaram a evidenciar que o modelo conceitual em OntoUML é um instrumento que pode ser inserido no processo de desenvolvimento de software, principalmente no contexto de Sistemas de Informação. O modelo conceitual pode contribuir com o entendimento mais real do domínio, antes de que artefatos tecnológicos pertinentes a Sistemas de Informação sejam desenvolvidos. Quanto maior for o entendimento do domínio, menor serão os problemas no desenvolvimento do software decorrente da falta de compreensão. A lista de requisitos gerada pelo método proposto permite uma visão geral do escopo do domínio e também uma visão geral dos requisitos necessários, caso um Sistema de Informação seja desenvolvido. A inserção da construção do modelo conceitual em OntoUML no processo de desenvolvimento de software é uma atividade que deve aumentar o tempo alocado para o levantamento de requisitos. Porém, acredita-se que os benefícios de uma compreensão mais abrangente do domínio no início do processo podem evitar vários problemas em etapas posteriores ao desenvolvimento do software.

REFERÊNCIAS DA REVISÃO SISTEMÁTICA

ID	Referência
1	Jureta IJ, Borgida A, Ernst N, Mylopoulos (2010) Techne: Towards a New Generation of Requirements Modeling Languages with Goals, Preferences, and Inconsistency Handling. In: International Requirements Engineering Conference pp 115–124
2	Garrido JL, Noguera M, González M, Hurtado MV, Rodríguez ML (2007) Definition and use of Computation Independent Models in an MDA-based groupware development process. <i>Science of Computer Programming</i> 66:25–43
3	Zhang H, Kishore R, Sharman R, Ramesh R (2007) Agile Integration Modeling Language (AIML): A conceptual modeling grammar for agile integrative business information systems. <i>Decision Support Systems</i> 44:266–284
4	Santana B, Diniz S, Barbosa J, Leite JCP (2008) A Language-Based Approach to Variability Analysis. In: Workshop on Requirements Engineering pp 179–190
5	Chen X, Yin B, Jin Z (2010) Dptool: A Tool for Supporting the Problem Description and Projection. International Requirements Engineering Conference pp 401–402
6	Hilaire V, Cossentino M, Gechter F, Rodriguez S, Koukam A (2013) An approach for the integration of swarm intelligence in MAS: An engineering perspective. <i>Expert Systems with Applications</i> 40:1323–1332
7	Cañete-Valdeón JM, Galán FJ, Toro M (2009) The intentional relationship of representation between the constructs of a language and reality. <i>Data & Knowledge Engineering</i> 68:173–191
8	Liu, C (2010) CDADE: Conflict detector in activity diagram evolution based on speech act and ontology. <i>Knowledge-Based Systems</i> 23:536–546
9	Ovaska E, Evesti A, Henttonen K, Palviainen M, Aho P (2010) Knowledge based quality-driven architecture design and evaluation. <i>Information and Software Technology</i> 52:577–601
10	Vongdoiwang W, Batanov DN (2006) An ontology-based procedure for generating object model from text description. <i>Knowledge and Information Systems</i> , 10(1):93–108
11	Pires PF, Delicato FC, Cóbe R, Batista T, Davis JG, Song JH (2011) Integrating ontologies, model driven, and CNL in a multi-viewed approach for requirements engineering. <i>Requirements Engineering</i> , 16:133-160 doi:10.1007/s00766-011-0116-1
12	Lindoso AN, Girardi R (2006) The SRAMO Technique for Analysis and Reuse of Requirements in Multi-agent Application Engineering. In Workshop on Requirements Engineering pp 1–10
13	Bimrah KK, Mouratidis H, Preston D (2008) Modelling Trust Requirements by Means of a Visualization Language. <i>Requirements Engineering Visualization</i> pp 26–30
14	Rajsiri V, Lorré JP, Bénaben F, Pingaud H (2010) Knowledge-based system for collaborative process specification. <i>Computers in Industry</i> 61:161–175
15	Beydoun G, Low G, Tran N, Bogg P (2011) Development of a peer-to-peer information sharing system using ontologies. <i>Expert Systems with Applications</i> . 38:9352–9364

16	Santos EG, Medeiros AP (2011) Design Rationale Representation in Requirements Engineering using the KAOS meta-model In: Workshop on Requirements Engineering
17	Bolloju N, Sugumaran V A (2012) Knowledge-based object modeling advisor for developing quality object models. Expert Systems with Applications 39:2893–2906
18	Bolloju N, Schneider C, Sugumaran V (2012) A knowledge-based system for improving the consistency between object models and use case narratives. Expert Systems with Applications 39:9398–9410
19	Fan S, Zhao JL, Dou W, Liu M (2012) A framework for transformation from conceptual to logical workflow models. Decision Support Systems 54:781–794
20	Ghidini C, Francescomarino C, Rospocher M, Tonella P, Serafini L (2012) Semantics-Based Aspect-Oriented Management of Exceptional Flows in Business Processes. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42:25–37
21	Wand Y, Monarchi DE, Parsons J, Woo CC (1995) Theoretical foundations for conceptual modelling in information systems development. Decision Support Systems 15:285–304
22	Green P, Rosemann M (2000) Integrated process modeling: an ontological evaluation. Information systems 25(2):73-87
23	Rosemann M, Green P (2002) Developing a meta model for the Bunge–Wand–Weber ontological constructs. Information Systems 27:75–91
24	Wagner G (2003) The Agent–Object–Relationship metamodel: towards a unified view of state and behavior. Information Systems 28:475–504
25	Evermann J, Wand Y (2005) Toward formalizing domain modeling semantics in language syntax. IEEE Transactions on Software Engineering 31:21–37
26	Gemino A, Wand Y (2005) Complexity and clarity in conceptual modeling: Comparison of mandatory and optional properties. Data & Knowledge Engineering 55:301–326
27	Etien A, Rolland C (2005) Measuring the fitness relationship. Requirements Engineering 10(3):184–197
28	Perez GC, Sellers, BH (2007) Modelling software development methodologies: A conceptual foundation. Journal of Systems and Software 80:1778–1796
29	Dreiling A, Rosemann M, Wil MPA, Sadiq W (2008) From conceptual process models to running systems: A holistic approach for the configuration of enterprise system processes. Decision Support Systems 45:189–207
30	Jureta IJ, Mylopoulos J, Faulkner S (2008) Revisiting the Core Ontology and Problem in Requirements Engineering. In: International Requirements Engineering Conference pp 71–80
31	Bera P, Evermann J (2012) Guidelines for using UML association classes and their effect on domain understanding in requirements engineering. Requirements Engineering doi:10.1007/s00766-012-0159-y
32	Oliveira KM, Zlot F, Rocha AR, Travassos GH, Galotta C, Menezes CS (2004) Domain-oriented software development environment. Journal of Systems and Software 72:145–161
33	Kilov H, Sack I (2009) Mechanisms for communication between business and IT experts. Computer Standards & Interfaces 31:98–109
34	Regoczei S, Plantinga EPO (1987) Creating the domain of discourse: ontology and inventory. International Journal of Man-Machine Studies.

	27:235–250
35	Jin Z (2003) Automatically multi-paradigm requirements modeling and analyzing: An ontology-based approach. <i>Science in China</i> 46(4):279-297 doi:10.1360/02yf0093
36	Kaiya H, Saeki M (2006) Using Domain Ontology as Domain Knowledge for Requirements Elicitation. In: <i>International Requirements Engineering Conference</i> pp 189–198
37	Girardi R, Marinho LB (2006) A domain model of Web recommender systems based on usage mining and collaborative filtering. <i>Requirements Engineering</i> 12(1):23–40
38	Breaux, T (2009) Exercising Due Diligence in Legal Requirements Acquisition: A Tool-supported, Frame-Based Approach. In: <i>International Requirements Engineering Conference</i> pp 225–230
39	Biletskiy Y, Ranganathan GR (2010) A semantic approach to a framework for business domain software systems. <i>Computers in Industry</i> 61:750–759
40	Tacla CA, Freddo AR, Paraiso EC, Ramos MP, Sato GY (2011) Supporting small teams in cooperatively building application domain models. <i>Expert Systems with Applications</i> 38:1160–1170
41	Bagheri E, Ensan F, Gasevic D (2012) Decision support for the software product line domain engineering lifecycle. <i>Automated Software Engineering</i> 19(3):335–377
42	Cossentino M, Gaud N, Hilaire V, Galland S, Koukam (2009) A ASPECS: an agent-oriented software process for engineering complex systems. <i>Autonomous Agents and Multi-Agent Systems</i> 20(2): 260–304
43	Aranda GN, Vizcaíno A, Piattini M (2010) A framework to improve communication during the requirements elicitation process in GSD projects. <i>Requirements Engineering</i> 15(4):397-417
44	Cysneiros LM, Kushniruk A (2003) Bringing Usability to the Early Stages of Software Development Usability Ontology References. In: <i>International Requirements Engineering Conference</i>
45	Kof L (2007) Scenarios: Identifying Missing Objects and Actions by Means of Computational Linguistics. In: <i>International Requirements Engineering Conference</i> pp 121–130
46	Katz S, Rashid A (2004) From aspectual requirements to proof obligations for aspect-oriented systems. <i>International Requirements Engineering Conference</i> pp 43–52
47	Masuwa-Morgan K, Burrell P (2004) Justification of the need for an ontology for accessibility requirements (Theoretic framework). <i>Interacting with Computers</i> 16:523–555
48	Cabral G, Sampaio A (2008) Formal Specification Generation from Requirement Documents. <i>Electronic Notes in Theoretical Computer Science</i> . 195:171–188
49	Weber-Jahnke JH, Onabajo A (2009) Finding Defects in Natural Language Confidentiality Requirements. In: <i>International Requirements Engineering Conference</i> pp 213–222
50	Rago A, Marcos C, Diaz-Pace JA (2011) Uncovering quality-attribute concerns in use case specifications via early aspect mining. <i>Requirements Engineering</i> , 18(1):67–84
51	Castañeda V, Ballejos L, Caliusco ML (2012) Improving the Quality of

	Software Requirements Specifications with Semantic Web Technologies. In Workshop on requirements engineering
52	Verlaine B, Dubois Y, Jureta IJ, Faulkner S (2012) Towards conceptual foundations for service-oriented requirements engineering: bridging requirements and services ontologies. IET Software. 6(2):85-102 doi: 10.1049/iet-sen.2011.0027
53	Richter H, Gandhi R, Liu L, Lee S, Carolina N (2006) Incorporating Multimedia Source Materials into a Traceability Framework. In: International Workshop on Multimedia Requirements Engineering p 5
54	Gandhi RA, Lee SW (2007) Discovering and Understanding Multi-dimensional Correlations among Certification Requirements with application to Risk Assessment. In: International Requirements Engineering Conference. 231–240 doi:10.1109/RE.2007.46
55	Zhang Y, Witte R, Rilling J, Haarslev V (2008) Ontological approach for the semantic recovery of traceability links between software artefacts. IET Software 2(3) pp 185-203 doi: 10.1049/iet-sen:20070062
56	Assawamekin N, Sunetnanta T, Pluempitiwiriyaewej C (2009) Ontology-based multiperspective requirements traceability framework. Knowledge and Information Systems 25(3):493–522
57	Chicaiza J, López J, Piedra N, Martínez O, Tovar E (2010) Usage of social and semantic web technologies to design a searching architecture for software requirement artifacts. IET Software 4(6): 407–417
58	Supakkul S, Chung L (2010) Visualizing non-functional requirements patterns. In: International Workshop on Requirements Engineering Visualization pp 25–34
59	Cappelli C, Leite JCSP, Oliveira APA (2007) Exploring Business Process Transparency Concepts. In: International Requirements Engineering Conference pp 389–390
60	Kaindl H, Svetinovic D (2010) On confusion between requirements and their representations. Requirements Engineering, 15(3):307–311

REFERÊNCIAS BIBLIOGRÁFICAS

- (ARANDA et al., 2008) ARANDA, G.N., VIZCAÍNO, A., CECHICH, A., PIATTINI, M.: **A Methodology for Reducing Geographical Dispersion Problems during Global Requirements Elicitation**. In: Workshop on Requirements Engineering, p. 117–127, 2008.
- (ARANGO, 1994) ARANGO, G. **Domain Analysis Methods**, Workshop on Software Architecture, Los Angeles: USC Center for Software Engineering, 1994.
- (ARPÍREZ et al., 2001) ARPÍREZ, J. C, CORCHO, O.; FERNANDEZ-LOPEZ, M.; GOMEZ-PEREZ, A. **WebODE: a scalable ontological engineering workbench**, in: First International Conference on Knowledge Capture (KCAP_01), ACM Press, Victoria, p. 6–13, 2001.
- (BARCELOS et al., 2011) BARCELOS, P. P. F.; GUIZZARDI, G., GARCIA, A. S.; MONTEIRO, M. E. **Ontological Evaluation of the ITU-T Recommendation G.805**. In 2011 18th International Conference on Telecommunications. IEEE.
- (BERA; EVERMANN, 2012) BERA, P.; EVERMANN, J. **Guidelines for using UML association classes and their effect on domain understanding in requirements engineering**. Requirements Engineering, 2012.
- (BERNARAS et al., 1996) BERNARAS, A.; LARESGOITI, I., CORERA; J. **Building and reusing ontologies for electrical network applications**, in: Proc. European Conference on Artificial Intelligence (ECAI_96), Budapest, Hungary, p. 298–302, 1996.
- (BOEHM, 1981) BOEHM, B. Software Engineering Economics, Prentice-Hall, 1981.
- (BORST, 1997) BORST W. N. **Construction of Engineering Ontologies**. University of Tweenty. Ensched, The Netherdands - Centre for Telemática and Information Technology, 1997.
- (BOUQUET et al., 2004) BOUQUET, P.; GIUNCHIGLIA, F.; VANHARMELEN F.; SERAFINI L.; STUCKENSCHMIDT H. **Contextualizing ontologies**. Web Semantics: Science, Services and Agents on the World Wide Web, v. 1, n. 4, p. 325-343, 2004.
- (CAÑETE-VALDEÓN et al., 2009) CAÑETE-VALDEÓN, J. M., GALÁN, F. J., TORO, M. **The intentional relationship of representation between the constructs of a language and reality**. Data & Knowledge Engineering, v. 68, p. 173–191, 2009.
- (CAPUCHINO et al., 2000) Capuchino, A.M.; Juristo, N.; Van de Riet, R.P. **Formal justification in object-oriented modelling: A linguistic approach**. Data & Knowledge Engineering, v.33, i.1, p. 25-47, 2000.
- (CASTRO, 1995) CASTRO, J.F. Introdução a Engenharia de Requisitos. XV Congresso da Sociedade Brasileira de Computação, JAI 95 – XIV Jornada de Atualização em Informática. Canela, RS – Brasil, 1995.

(CASTRO, 2010) CASTRO, L. **Abordagem Linguística para Modelagem Conceitual de Dados com Foco Semântico**. Dissertação de mestrado, Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, , Brasil, 2010.

(CORCHO et al., 2003) CORCHO, O.; LOPEZ, M. F.; PEREZ, A. G. **Methodologies, tools and languages for building ontologies. Where is their meeting point?** Data & Knowledge Engineering, v. 46, n. 1, p. 41-64, 2003.

(DEAN et al., 2002) DEAN, M.; CONNOLLY, D.; HARMELEN, F.; HENDLER, J.; HORROCKS, I.; MCGUINNESS, D.L.; PATEL-SCHNEIDER, P.F.; STEIN, L.A. **OWL Web Ontology Language 1.0 Reference**, W3C Working Draft, 2002. Disponível em: <<http://www.w3.org/TR/owl-ref/>>. Acesso em: Maio, 2011.

(DEEPTIMAHANTI & SANYAL, 2011) D. K. DEEPTIMAHANTI AND R. SANYAL. **Semi-automatic generation of UML models from natural language requirements**. In Proceedings of the 4th India Software Engineering Conference. ACM, 165–174, 2011.

(DIXON, 2005) DIXON, R. M. **A Semantic Approach to English Grammar**, 2nd ed. Oxford University Press, USA, 2005.

(DOMINGUE, 1998) DOMINGUE, J. **Tadzebao and Webonto: Discussing, Browsing and Editing Ontologies on the Web**, in: Proc. 11th Knowledge Acquisition Workshop (KAW98), Banff, 1998.

(EVERMANN; WAND, 2005) EVERMANN, J., WAND, Y. **Toward formalizing domain modeling semantics in language syntax**. IEEE Transactions on Software Engineering. v. 31, p. 21–37, 2005.

(FALBO et al., 1998) FALBO, R. A., MENEZES, C. S., ROCHA, A. R. C. **A Systematic Approach for Building Ontologies**, In Proceedings of the Sixth Iberoamerican Conference on Artificial Intelligence (IBERAMIA), Lisboa, Portugal, 1998.

(FAWCETT, 2006) FAWCETT, T. An introduction to ROC analysis. Pattern Recognition Letters, 27, 861–874, 2006.

(FRANCO; LEITE, 1992) FRANCO, A. P. M.; LEITE, J.C.S.P. **Uma estratégia de Suporte a Engenharia de Requisitos**, In Anais do XIX Seminario Integrado de Software y Hardware, Rio de Janeiro, pp, 200-213, 1992.

(GANDHI; LEE, 2007) GANDHI, R. A.; LEE, S. W. **Discovering and Understanding Multi-dimensional Correlations among Certification Requirements with application to Risk Assessment**. In: International Requirements Engineering Conference, p. 231–240, 2007.

(GAO, 1992) GAO US General Accounting Office, **Mission Critical Systems: Defense Attempting to Address Major Software Challenges**, GAO/IMTEC-93-13, December 1992

(GARRIDO et al., 2007) GARRIDO, J. L., NOGUERA, M., GONZÁLEZ, M., HURTADO, M. V., RODRÍGUEZ, M. L. **Definition and use of Computation**

Independent Models in an MDA-based groupware development process. Science of Computer Programming, v. 66, p. 25–43, 2007.

(GEMINO; WAND, 2005) GEMINO, A., WAND, Y. **Complexity and clarity in conceptual modeling: Comparison of mandatory and optional properties.** Data & Knowledge Engineering, v. 55, p. 301–326, 2005.

(GENESERETH; FIKES, 1992) GENESERETH, M.; FIKES, R. **Knowledge interchange format**, Technical Report Logic-92-1, Computer Science Department, Stanford University, 1992.

(GHIDINI et al., 2012) GHIDINI, C.; FRANCESCO MARINO, C.; ROSPOCHER, M.; TONELLA, P.; SERAFINI, L. **Semantics-Based Aspect-Oriented Management of Exceptional Flows in Business Processes.** IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), v. 42, p. 25–37, 2012.

(GIL, 2002) GIL, A. C. **Como elaborar projetos de pesquisa.** 4. ed. São Paulo:Atlas, 2006. 175 p.

(GIRARDI; LEITE, 2008) GIRARDI, R.; LEITE, A. **A knowledge-based tool for multi-agent domain engineering.** Knowledge-Based Systems, v. 21, p. 604–611, 2008.

(GIRARDI; MARINHO, 2006) GIRARDI R.; MARINHO L.B. **A domain model of Web recommender systems based on usage mining and collaborative filtering.** Requirements Engineering, v. 12, n. 1, p. 23–40, 2006.

(GONÇALVES et al., 2007) GONÇALVES, B.; GUIZZARDI, G.; FILHO, J. G. P. An electrocardiogram (ECG) domain ontology. In 2nd Workshop on Ontologies and Metamodels for Software and Data Engineering, 2007.

(GRUBER, 1993) GRUBER, T. R. **Toward Principles for the Design of Ontologies Used for Knowledge Sharing.** International Journal of Human and Computer Studies, v. 43, n. 5-6, p. 907-928, 1993.

(GRUBER, 1993A) GRUBER, T. R. **A Translation Approach to Portable Ontology Specifications.** Knowledge Creation Diffusion Utilization, v. 5, n. 2, p. 199-220, 1993.

(GRUNINGER; FOX, 1995) GRUNINGER, M.; FOX, M.S. **Methodology for the design and evaluation of ontologies**, in: Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, 1995.

(GUARINO, 1998) GUARINO, N. **Formal Ontology and Information Systems.** In the Proceedings of Formal Ontology in Information Systems, Washington, DC: IOS Press, p. 3-15, 1998.

(GUIZZARDI, 2005) GUIZZARDI, G. **Ontological Foundations for Structural Conceptual Models**, Telematica Instituut Fundamental Research Series 15, Universal Press, 2005.

(GUIZZARDI, 2007) GUIZZARDI, G. **On ontology, ontologies, conceptualizations, modeling languages, and (meta) models**. *Frontiers in artificial intelligence and applications*, p. 18–28, 2007.

(GUIZZARDI et al., 2010) GUIZZARDI, G., BAIÃO, F., LOPES, M.; FALBO, R. **The Role of Foundational Ontologies for Domain Ontology Engineering: An Industrial Case Study in the Domain of Oil and Gas Exploration and Production**. *International Journal of Information System Modeling and Design*, v. 1, n. 2, p. 1–22.

(GUIZZARDI et al., 2011) GUIZZARDI, G., DAS GRAÇAS, A., GUIZZARDI, R. S. S., **Design Patterns and Inductive Modeling Rules to Support the Construction of Ontologically Well-Founded Conceptual Models in OntoUML**, 3rd Intl. Workshop on Ontology Driven Inf. Systems (ODISE 2011), London.

(GUIZZARDI et al., 2012) GUIZZARDI, RENATA, FRANCH, XAVIER, GUIZZARDI, GIANCARLO. **Applying a Foundational Ontology to Analyze Means-End Links in the i* Framework**. In *Proceedings of the 6th International Conference on Research Challenges in Information Science*, Valencia, Spain. 2012

(HARINEK; ŠIMKOM, 2013) HARINEK, J.; ŠIMKOM M. **Improving term extraction by utilizing user annotations**. *Proceedings of the 2013 ACM symposium on Document engineering - DocEng*, 2013.

(HARMAIN & GAIZAUSKAS, 2003) HARMAIN, H. M., GAIZAUSKAS R. **CM-Builder: A Natural Language-Based CASE Tool for Object- Oriented Analysis**. *Automated Software Engineering.*, v. 10(2), p.157-181, 2003.

(HARZALLAH et al., 2012) HARZALLAH, M., BERIO, G., OPDAHL, A. L.: **New perspectives in ontological analysis: Guidelines and rules for incorporating modelling languages into UEML**. *Information Systems*, v. 37, p. 484–507, 2012.

(HENDERSON-SELLERS et al., 2015) Henderson-Sellers, B., Gonzalez-Perez, C., Eriksson, O., Ågerfalk, P.J. **Software modeling languages: a wish list**. In *Seventh International Workshop on Modeling in Software Engineering*, pp 72-77, 2015.

(HICKEY; DAVIS, 2003) A. M. HICKEY, A. M. DAVIS: **Elicitation Technique Selection: How Do Experts Do it?** *Proceeding of the 11th IEEE International Requirement Engineering Conference*, Monterey Bay, USA, p. 169-178, 2003.

(HORROCKS et al., 2000) HORROCKS, I.; FENSEL, D.; HARMELEN, F.; DECKER, S.; ERDMANN, M.; KLEIN M., **OIL in a Nutshell**, in: *ECAI_00 Workshop on Application of Ontologies and PSMs*, Berlin, 2000.

(JACOBSON, 1992) JACOBSON, I. **Object-Oriented Software Engineering**, Addison-Wesley, 1992.

(JURETA et al., 2010) JURETA, I. J.; BORGIDA, A.; ERNST, N. A.; MYLOPOULOS, J. **TECHNE: Towards a New Generation of Requirements Modeling Languages with Goals, Preferences, and Inconsistency Handling**. *International Requirements Engineering Conference*, p. 115–124, 2010.

(KILOV; SACK, 2009) KILOV, H., SACK, I. **Mechanisms for communication between business and IT experts**. Computer Standards & Interfaces, v. 31, p. 98–109, 2009.

(KITCHENHAM et al., 2004) KITCHENHAM, B.A.; TRAVASSOS, G.H.; MAYRHAUSER, A.; NIESSINK, F.; SCHNEIDEWIND, N.F.; SINGER, J.; TAKADA, S.; VEHVILAINEN, R.; YANG, H. **Towards an Ontology of Software Maintenance**. Journal of Software Maintenance: Research and Practice, p. 365–389, 1999.

(LACY, 2005) LACY, L. W. **OWL: Representing Information Using the Web Ontology Language**, Trafford Publishing, 2005.

(LASSILA; MCGUINNESS, 2001) LASSILA, O., MCGUINNESS, D. L. **The Role of Frame-Based Representation on the Semantic Web**, Knowledge Systems Laboratory, Report KSL-01-02, January, 2001.

(LEÃO et al., 2012) LEÃO, F.; DIIRR, T.; BAIÃO, F.; REVOREDO, K. **Construção de Modelos Conceituais a Partir de Textos com Apoio de Tipos Semânticos**. In Proceedings of Joint V Seminar on Ontology Research in Brazil and VII International Workshop on Metamodels, Ontologies and Semantic Technologies, Recife, Brazil, Vol-938, 2012.

(LEÃO et al., 2013) LEÃO, F.; REVOREDO, K.; BAIÃO, F. **Learning Well-Founded Ontologies Through Word Sense Disambiguation**. In Proceeding Of: 2nd Brazilian Conference On Intelligent Systems (Bracis-13), 2013.

(LEÃO, F., 2014) LEÃO, F. **Expanding the Semantic Knowledge of Wordnet Through Semantic Types and UFO**. Dissertação de mestrado. Universidade Federal do Estado do Rio de Janeiro. Rio de Janeiro, setembro de 2014.

(LEÃO, F., 2016) LEÃO, F. **LEO - Learning Ontologies System**. Acesso: 01/11/2016, disponível em: <https://github.com/felipeleao/leo>.

(LEE; GANDHI, 2005) LEE, S. W., GANDHI, R. A. **Ontology-based active requirements engineering framework**. Engineering Conference, 2005. APSEC' 8 pp, 2005.

(LEITE et al., 1997) LEITE J. C. S. P.; ROSSI G.; et al. **Enhancing a Requirements Baseline with Scenarios**, Proceedings of RE 97': International Symposium on Requeriments Engineering, IEEE, January 1997.

(LI, 2005) LI, L. **Ontological modeling for software application development**. Advances in Engineering Software, v. 36, p. 147–157, 2005.

(LOUCOPOULOS; KARAKOSTAS, 1995) LOUCOPOULOS, P., KARAKOSTAS, V. **System Requirements Engineering**. McGraw-Hill, 1995.

(LUIS et al., 2008) LUIS J.; VARA D.; SÁNCHEZ J. **Improving Requirements Analysis through Business Process Modelling: a Participative Approach**, v. 1, p. 165–176.

(MACEDO; LEITE, 1999) MACEDO, N.; LEITE, J. **Elicit@99: um protótipo de ferramenta para a elicitação de requisitos.** In Workshop em Engenharia de Requisitos, 1999.

(MARTINS; DALTRINI, 1999) MARTINS, L. E. G; DALTRINI, B. M. **Activity Theory: a Framework to Software Requirements Elicitation.** In Workshop em Engenharia de Requisitos, 1999.

(MAYR & KOP, 2002) MAYR, H.C.; KOP, C. **A user centered approach to requirements modeling.** In: M.Glinz, G. Müller-Luschnat (eds.): Proc. "Modellierung 2002". Lecture Notes in Informatics P-12 (LNI), GI-Edition, p.75-86, 2002.

(MUSEN, 2002) MUSEN, M. A. **Medical Informatics: Searching for Underlying Components.** Methods of information in medicine, v. 41, n. 1, p. 12–19, 2002.

(MYLOPOULOS, 1992) MYLOPOULOS, J. **Conceptual modeling and Telos,** In P. Loucopoulos and R. Zicari, editors, Conceptual modeling, databases, and CASE. Wiley, 1992.

(NOY; MCGUINNESS, 2001) NOY, N. F., MCGUINNESS, D. L. **Ontology Development 101: A Guide to Creating your First Ontology,** Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March, 2001.

(OLIVEIRA et al., 2004) OLIVEIRA, K. M.; ZLOT, F.; ROCHA, A. R.; TRAVASSOS, G. H.; GALOTTA, C.; MENEZES, C. S. **Domain-oriented software development environment.** Journal of Systems and Software, v. 72, p. 145–161, 2004.

(OLED, 2014) <https://code.google.com/p/OntoUML-lightweight-editor/>, acessado em: 2014 fev 11.

(PEDERSEN; KOLHATKAR, 2009) PEDERSEN, T.; KOLHATKAR, V. **WordNet:: Senses relate:: All words: A Broad Coverage Word Sense Tagger That Maximizes Semantic Relatedness.** Human Language Technologies, 2009.

(PEREZ et al., 1996) PEREZ, A.G.; LOPEZ, M.; VICENTE, A. **Towards a Method to conceptualize domain ontologies.** In: Ecai96 Workshop on Ontological Engineering, Budapest, p. 41–51, 1996.

(PIRES et al., 2011) PIRES, P. F.; DELICATO, F. C.; CÔBE, R.; BATISTA, T.; DAVIS, J. G.; SONG, J. H. **Integrating ontologies, model driven, and CNL in a multi-viewed approach for requirements engineering.** Requirements Engineering, v. 16, n. 2, 2011.

(POHL, 1997) POHL, K. **Requirements engineering: An overview.** In Encyclopedia of Computer Science and Technology. A. Kent, and J. Williams, Eds. Marcel Dekker, New York, NY, v. 36, suppl. 21, 1997.

(POHL; RUPP, 2011) POHL, K.; RUPP, C. **Requirements engineering fundamentals: A study guide for the Certified Professional for Requirements Engineering exam: Foundation level, IREB compliant, 1º ed.,** Rocky Nook Inc., 2011.

(PROTÉGÉ, 2011) **The Protégé Ontology Editor and Knowledge Acquisition System**. Disponível em: <<http://protege.stanford.edu/index.html>>, Acesso em 05 abr. 2014.

(ROBINSON; PAWLOWSKI, 1999) ROBINSON, W., PAWLOWSKI, S. **Managing requirements inconsistency with development goal monitors**. IEEE Transactions on Software Engineering, 25, 1999.

(ROSEMANN, GREEN, 2002) ROSEMANN, M.; GREEN, P. **Developing a meta model for the Bunge–Wand–Weber ontological constructs**. Information Systems, v. 27, p. 75–91, 2002.

(SANTANDER; CASTRO, 2000) SANTANDER, V.; CASTRO, J. **Desenvolvendo Use Cases a partir de Modelagem Organizacional**, In Workshop em Engenharia de Requisito, p.158-180, 2000.

(SURE et al., 2002) SURE, Y.; ERDMANN, M.; ANGELE, J.; STAAB, S.; STUDER, R.; WENKE, D. **OntoEdit: collaborative ontology engineering for the semantic web**, in: First International Semantic Web Conference (ISWC_02), Lecture Notes in Computer Science, v. 2342, Springer, Berlin, p. 221–235, 2002.

(STUDER et al., 1998) STUDER, R.; BENJAMINS, V. R.; FENSEL, D. **Knowledge engineering: principles and methods**. Data & Knowledge Engineering, v. 25, n. 1-2, p. 161-197, 1998.

(SWARTOUT et al., 1997) SWARTOUT, B., RAMESH P., KNIGHT, K., RUSS, T. **Toward Distributed Use of Large-Scale Ontologies**. In: AAAI Symposium on Ontological Engineering, Stanford, 1997.

(THAYER; DORFMAN, 1997) THAYER, R. H.; DORFMAN, M. **Software Requirements Engineering**, 2d Ed. IEEE Computer Society Press, 1997.

(TOPIA, 2013) **Topia.termextract 1.1.0**. Disponível em: <<https://pypi.python.org/pypi/topia.termextract>>, Acesso em 01 nov 2013.

(USCHOLD; KING, 1995) USCHOLD, M.; KING, M. Towards a Methodology for Building Ontologies, in: IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, 1995.

(UML, 2014) UML – Unified Modeling Language. Disponível em <<http://www.uml.org/>>, Acesso em 2014 fev 11.

(VALASKI et al., 2013a) VALASKI, J.; STANCKE, W.; REINEHR S.; MALUCELLI, A. **Retrospective and Trends in Requirements Engineering through the WER**. In: Workshop em Engenharia de Requisitos (WER), Montevideo, 2013.

(VALASKI et al., 2013b) VALASKI, J.; STANCKE, W.; REINEHR S.; MALUCELLI, A. **WER Overview: Retrospective, Trends and Relevance**. Clei Electronic Journal, v.10, n. 3, paper 3, 2013.

(VALASKI et al., 2013c) VALASKI, J.; STANCKE, W.; REINEHR S.; MALUCELLI, A. **Apoio Semântico à Engenharia de Requisitos**. In Anais Proceedings of

Requirements Engineering@Brazil(ER@BR 2013), Rio de Janeiro, Brazil, July 16, 2013.

(VALASKI et al., 2014) VALASKI, J.; REINEHR S.; MALUCELLI, A. **Environment for Requirements Elicitation Supported by Ontology-Based Conceptual Models: A Proposal**. In Proceedings of the 2014 International Conference on Software Engineering Research and Practice (SERP'14), ISBN 1-60132-286-0, Las Vegas, USA, p. 144-150, 2014.

(VALASKI et al., 2015) VALASKI, J.; REINEHR S.; MALUCELLI, A. **Approaches and Strategies to Extract Relevant Terms: How are they being applied?**. In: The 17th International Conference on Artificial Intelligence, 2015, Las Vegas. Int'l Conf. Artificial Intelligence ICAI'2015. Las Vegas: CSREA Press, 2015. p. 478-484.

(VALASKI et al., 2016a) VALASKI, J.; REINHER S.; MALUCELLI, A. **Which Roles Ontologies play on Software Requirements Engineering? A Systematic Review**. In: The 2016 International Conference on Software Engineering Research & Practice, 2016, Las Vegas. Proceedings of the 2016 International Conference on Software Engineering Research & Practice, 2016. p. 24-30.

(VALASKI et al., 2016b) VALASKI, J.; REINHER S.; MALUCELLI, A. **Evaluating the Expressiveness of a Conceptual Model Represented in OntoUML and UML**. In: ONTOBRAS - Brazilian Ontology Research Seminar, 2016, Curitiba. ONTOBRAS, 2016.

(VALASKI et al., 2016c) VALASKI, J.; REINHER S.; MALUCELLI, A. **Computational Environment to Semi-Automatically Build a Conceptual Model Represented in OntoUML**. In: ONTOBRAS - Brazilian Ontology Research Seminar, 2016, Curitiba. ONTOBRAS, 2016.

(VALASKI et al., 2017) VALASKI, J.; REINHER S.; MALUCELLI, A. **Deriving Domain Functional Requirements from Conceptual Model Represented in OntoUML**. In Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 2, pages 263-270.

(WAGNER, 2003) WAGNER, G. **The Agent-Object-Relationship metamodel: towards a unified view of state and behavior**. Information Systems, v. 28, p. 475-504, 2003.

(WAND et al., 1995) WAND, Y.; MONARCHI, D. E.; PARSONS, J.; WOO, C. C. **Theoretical foundations for conceptual modelling in information systems development**. Decision Support Systems, v. 15, p. 285-304, 1995.

(WEBEDITOR, 2014) Site: <https://code.google.com/p/OntoUMLwebeditor/>, acessado em: 2014 fev 11.

(WORDNET.Pr, 2014) **WordNet A Lexical Database for English**. Disponível em: <<http://WordNet.princeton.edu/WordNet>>, Acesso em 04 abr. 2014.

(WORDNET.PT, 2014) **WordNet.PT Rede Léxico-Conceptual do Português**. Disponível em: <<http://www.clul.ul.pt/clg/WordNetpt/index.html>>, Acesso em 04 abr. 2014.

(WORDNET.Br, 2014) WordNet.Br 1-0 – Base de Verbos. Disponível em: <http://www.nilc.icmc.usp.br/WordNetbr>, Acesso em: 14 abr. 2014.

(YU, 1995) YU, ERIC. **Modelling Strategic Relationships for Process Reengineering**, Phd Thesis, University of Toronto, 1995.

(ZANLORENCI; BURNETT, 1998) ZANLORENCI, E. P.; BURNETT, R. C. **Modelo para Qualificação da Fonte de Informação do Cliente e de Requisito Funcional**, In Workshop em Engenharia de Requisitos, 1998.

(ZHANG et al., 2007) ZHANG, H.; KISHORE, R.; SHARMAN, R.; RAMESH, R. **Agile Integration Modeling Language (AIML): A conceptual modeling grammar for agile integrative business information systems**. Decision Support Systems, v. 44, p. 266 – 284, 2007.

APÊNDICE A – INSTRUMENTO PARA AVALIAR A EXPRESSIVIDADE DA ONTOUML E UML

Nome: _____

Leia abaixo as afirmações extraídas do domínio de **Procuração Eletrônica**, analise a representação correspondente nos modelos conceituais (Anexo 1 - OntoUML e Anexo 2 - UML) e assinale com um **X** o modelo que **melhor representa** (está mais claro) o que está afirmado.

1. Uma Organização pode ter 1 ou vários representantes.

Opção 1 - No modelo OntoUML está mais claro.	
Opção 2 - No modelo UML está mais claro.	
Opção 3 - Ambos os modelos possuem o mesmo nível de clareza	

2. Um Representante é uma pessoa que pode representar 1 ou várias organizações.

Opção 1 - No modelo OntoUML está mais claro.	
Opção 2 - No modelo UML está mais claro.	
Opção 3 - Ambos os modelos possuem o mesmo nível de clareza	

3. Uma **Representação** é uma relação estabelecida entre 1 organização e 1 ou vários representantes.

.....

4. **Organização base de dados ICMS** é uma organização que tem registro na base de dados do ICMS.
5. **Usuário Receita** é uma pessoa que tem registro na base de dados da receita.
6. **Procurador** é um usuário ativo na receita que recebe 1 ou várias procurações.
7. **Concessor** é um representante de uma organização que concede 1 ou várias procurações.
8. A relação **Procuração** entre o mesmo concessor e o mesmo procurador pode ocorrer apenas 1 vez.
9. O **Concessor** da procuração deve ser o representante da organização relacionada à procuração.
10. **Organização Procuração** é a organização relacionada à procuração.
11. **Organização Procuração** tem registro na base de dados do ICMS.
12. Uma **Procuração** é uma relação estabelecida entre 1 concessor, 1 procurador, 1 organização e 1 ou vários serviços.

APÊNDICE B - INSTRUMENTOS PARA AVALIAR A IDENTIFICAÇÃO AUTOMÁTICA DOS CONSTRUTOS ONTOUML

INSTRUMENTO 1

1 - Há quantos anos desenvolve pesquisas utilizando a linguagem OntoUML?	
2 - A qual instituição de ensino/pesquisa está vinculado?	
3 - Qual é o seu nível de conhecimento da linguagem OntoUML?	

TEXTO 1 - SALE COMMISSION

Vendors may be employees or companies. Employees receive a salary amount and a commission amount, whereas companies only receive a commission amount. Each order corresponds to one vendor only, and each vendor has made at least one order, which is identified by an order number. The same salary may be paid to different sales employees. The same commission amount may be paid to different companies and to different sales employees. A monthly payment is made to all vendors. When a vendor makes a sale, he/she reports the order to the system. The system then confirms the order to the customer, and orders are delivered to customers weekly.

Conceito	Construto OntoUML	Este conceito é relevante para o modelo conceitual?	Este construto é o mais adequado para a representação do conceito?	Caso negativo, qual é o construto mais adequado?
commission amount	datatype			
company	kind			
customer	role			
employee	role			
order	kind			
order number	datatype			
payment	relator			
sale	relator			
sale employee	association			
vendor	role			
salary	datatype			
salary amount	datatype			
De acordo com o texto, insira abaixo termos que considera que devam ser incluídos no modelo conceitual e seu respectivo				

TEXTO 2 - LIBRARY

A library issues loan items to customers. Each customer is known as a member and is issued a membership card that shows a unique member number. Along with the membership number, other details on a customer must be kept such as a name, address, and date of birth. The library is made up of a number of subject sections. Each section is denoted by a classification mark. A loan item is uniquely identified by a bar code. There are two types of loan items, language tapes, and books. A language tape has a title language, and level. A book has a title, and author(s). A customer may borrow up to a maximum of 8 items. An item can be borrowed, reserved or renewed to extend a current loan. When an item is issued the customer's membership number is scanned via a bar code reader or entered manually. If the membership is still valid and the number of items on loan less than 8, the book bar code is read, either via the bar code reader or entered manually. If the item can be issued (e.g., not reserved) the item is stamped and then issued. The library must support the facility for an item to be searched and for a daily update of records.

Conceito	Construto OntoUML	Este conceito é relevante para o modelo conceitual?	Este construto é o mais adequado para a representação do conceito?	Caso negativo, qual é o construto mais adequado?
author	role			
book	kind			
customer	role			
item	kind			
loan item	association			
library	kind			
loan	relator			
member	role			
membership card	kind			
section	mode			
language tape	-			
bar code	datatype			
book bar code	datatype			
classification mark	-			
detail	datatype			
member number	datatype			
membership number	datatype			
number	datatype			
title	datatype			
title language	-			
De acordo com o texto, insira abaixo termos que considera que devam ser incluídos no modelo conceitual e seu respectivo construto! Insira novas linhas se necessário.				

TEXTO 3 - CANDIDATE				
<p>Candidate will register with system to hire services. Candidate can provide information about his academic, work experience and referee details. Candidates may make verification request during registration or some time later. Candidate updates his details at any time. System will inform service seeker about approximate time period required to provide service. System records details of candidate. System shall keep status of each request up-to-date. System shall interact with education institute systems to verify originality of degree. System asks type of service required. Service may be Standard, Silver or Gold. System enters request as a record in system. System informs about outcome to the candidate. Referees send recommendation letters using system on behalf of students. Customers pay fee for each type of service. Customer may be candidate or employer. Standard service verifies details of education. Silver service verifies details of education and profession. Gold service verifies details of education, profession and recommendation letters from referees. Employer accesses system to hire services of system for verification purposes. Employer registers with system. After, employer should provide system with information on type of service required along with candidate unique identification number. System records details of employer. Verification officer accesses system to retrieve verification and inquiry requests. Verification officer performs all types of verifications requested by candidate and employers upon retrieving requests. Verification officer also verifies authenticity of referee. Verification officer uses system to send to referees a request for a recommendation letter when candidate requests gold service.</p>				
Conceito	Construto OntoUML	Este conceito é relevante para o modelo conceitual?	Este construto é o mais adequado para a representação do conceito?	Caso negativo, qual é o construto mais adequado?
candidate	role			
customer	role			
employer	role			
education	mode			
institute	kind			
gold service	-			
fee	datatype			
profession	relator			
experience	mode			
recommendation letter	-			
referee	role			
request	relator			
service	relator			
silver service	-			
student	role			
type	mode			
verification officer	role			
authenticity	datatype			
behalf	relator			
candidate requests gold	-			
detail	datatype			
identification number	datatype			
time period	datatype			
verification	relator			
verification purposes	-			
De acordo com o texto, insira abaixo termos que considera que devam ser incluídos no modelo conceitual e seu respectivo construto! Insira novas linhas se necessário.				

INSTRUMENTO 2

	1 - Há quantos anos desenvolve pesquisas utilizando a linguagem OntoUML?			
	2 - A qual instituição de ensino/pesquisa está vinculado?			
	3 - Qual é o seu nível de conhecimento da linguagem OntoUML?			
TEXTO 1 - PUBLISHER				
Publishers publish books. Authors write books. A book can be written by several authors. An author has a unique name, a birth date and an address. A book has a unique title and a prize. An ISBN number identifies a book. Each book is published by exactly one publisher. A publisher has a unique denomination and an address. A publisher has employees. An employee has a unique social insurance number, a name and a birth date.				
Conceito	Construto OntoUML	Este conceito é relevante para o modelo conceitual?	Este construto é o mais adequado para a representação do conceito?	Caso negativo, qual é o construto mais adequado?
address	datatype			
author	role			
book	kind			
employee	role			
publisher	role			
denomination	datatype			
title	datatype			
birth date	datatype			
insurance number	datatype			
isbn number	datatype			
name	datatype			
price	datatype			
De acordo com o texto, insira abaixo termos que considera que devam ser incluídos no modelo conceitual e seu respectivo construto! Insira novas linhas se necessário.				
TEXTO 2 - CONFERENCE				
An Organizing Committee is composed by Committee Members and is responsible of organizing the Conference. The presiding Member of the Organizing Committee is the Chair, who publishes the Conference. Authors perform the submission of Papers, which are received by Committee Members who distribute them among Reviewers who are responsible for conducting reviews and recommending which papers should be accepted. A Paper can have three different stages during its lifecycle, before the reviews it is marked as a "Not Evaluated Paper", after the review, it becomes either an "Accepted Paper" or a "Rejected Paper". Presentations are prepared by Authors based on Accepted Papers only.				
Conceito	Construto OntoUML	Este conceito é relevante para o modelo conceitual?	Este construto é o mais adequado para a representação do conceito?	Caso negativo, qual é o construto mais adequado?
accepted paper	-			
author	role			
chair	role			
conference	kind			
committee members	-			
organizing committee	-			
paper	kind			
presentation	relator			
rejected paper	-			
reviewer	role			
review	relator			
submission	relator			
member	role			
stage	datatype			
De acordo com o texto, insira abaixo termos que considera que devam ser incluídos no modelo conceitual e seu respectivo construto! Insira novas linhas se necessário.				

TEXTO 3 - ROUTE BUS

There are two ways for people to travel with Voyager. Either passengers can make a reservation on a trip, or passengers can show up at the boarding gate without a reservation and purchase a ticket for an unreserved seat. Passengers with a reservation are assigned a reservation date, whereas, passengers without reservations are assigned a boarding date. The name and addresses of all passengers are collected. Telephone numbers are collected where possible. All bus trips are organized into daily route segments. All daily route segments have both a start time and an end time. Each daily route segment. Voyager organizes is classified as a route segment with a segment number, start town, and finish town. Voyager offers a range of trips, and each trip is made up of one or more route segments. For every trip there is a trip number, start town, and finish town. If the trip is organized around a special event, the event name is also associated with the trip. Each daily route segment that Voyager offers is part of a dally trip. A daily trip is undertaken by one or more bus drivers. The name, address, and employee number of all drivers is collected. Voyager also records information about absent drivers. When a driver is absent. Voyager records the absence start date and the details about the absence. The absent driver provides one or more reasons for being absent and each reason is assigned a detail number and a short description. Voyager also collects information about the buses used for daily trips. Buses have a make, model, and registration number. For buses in use, the average daily kilometers is collected. If a bus requires maintenance, Voyager notes the date on which the bus entered maintenance and records the one or more problems with the bus. Voyager assigns a problem number and a short description for every maintenance problem. Finally, the average cost to repair all problems with a bus in maintenance is also recorded.

Conceito	Construto OntoUML	Este conceito é relevante para o modelo conceitual?	Este construto é o mais adequado para a representação do conceito?	Caso negativo, qual é o construto mais adequado?
address	datatype			
bus	kind			
bus driver	role			
bus trips	association			
driver	role			
finish town	não identificado			
maintenance	relator			
maintenance problem	association			
make	kind			
model	kind			
passenger	role			
problem	não identificado			
reservation	relator			
route segment	não identificado			
start town	não identificado			
trip	relator			
absence	datatype			
absence start date	datatype			
cost	datatype			
date	datatype			
description	datatype			
detail	datatype			
employee number	datatype			
end time	datatype			
event name	datatype			
gate	kind			
information	não identificado			
kilometer	datatype			
name	datatype			
number	datatype			
problem number	datatype			
range	não identificado			
reason	não identificado			
registration_number	datatype			
reservation date	datatype			
segment number	datatype			
start time	datatype			
telephone number	datatype			
De acordo com o texto, insira abaixo termos que considera que devam ser incluídos no modelo conceitual e seu respectivo construto! Insira novas linhas se necessário.				

APÊNDICE C - INSTRUMENTOS PARA AVALIAR A DERIVAÇÃO DE REQUISITOS FUNCIONAIS DE DOMÍNIO

INSTRUMENTO 1

EXPERIMENTO PARA AVALIAR A COBERTURA E UTILIDADE DE UM MÉTODO PARA EXTRAÇÃO DE REQUISITOS FUNCIONAIS DE DOMÍNIO

INFORMAÇÃO DO PARTICIPANTE

Pergunta	Resposta do Participante
1. Hora início do experimento	
2. Nome	
3. Cargo/função profissional	
4. Curso de formação na graduação	
5. Nível de formação na área	Graduação <input type="checkbox"/> Especialização <input type="checkbox"/> Mestrado <input type="checkbox"/> Doutorado <input type="checkbox"/>
6. Já atuou/atua recebendo os requisitos prontos para o desenvolvimento do software	Sim <input type="checkbox"/> Não <input type="checkbox"/>
7. Se a resposta a questão 6 for Sim , por quantos anos	
8. Já atuou/atua diretamente com o Levantamento (Descoberta) de Requisitos de Software	Sim <input type="checkbox"/> Não <input type="checkbox"/>
9. Se a resposta a questão 8 for Sim , por quantos anos	
10. Com relação ao Levantamento de Requisitos de Software, qual é o seu grau de experiência?	(1) Sem experiência (2) Pouco experiente (3) Razoavelmente experiente (4) Experiente (5) Muito experiente

EXPLICAÇÃO DO EXPERIMENTO

- Este experimento trata da avaliação de um método que extrai Requisitos Funcionais a partir de textos.
- Os Requisitos Funcionais são extraídos em alto nível e são apresentados como possíveis candidatos a um requisito funcional em um Sistema de Informação.
- O método lista os Requisitos Funcionais de forma hierárquica considerando as dependências e as relações entre os conceitos identificados pelo método.
- A seguir são apresentadas as descrições de alguns domínios/negócios, a partir dos quais você poderia extrair uma lista inicial dos possíveis Requisitos Funcionais de um Sistema de Informação. Considerando apenas o que está descrito você deverá:
 - 1) Avaliar a qualidade do texto (respondendo 3 questões)
 - 2) Avaliar a cobertura dos Requisitos Funcionais gerados pelo método

EXECUÇÃO DO EXPERIMENTO

Texto 1 – Comissão de Vendas

Vendedores podem ser funcionários ou empresas. Funcionários recebem salário e comissão, enquanto que empresas recebem somente comissão. Cada ordem corresponde a somente um vendedor, e cada vendedor tem pelo menos uma ordem, a qual é identificada pelo número da ordem. O mesmo salário pode ser pago para diferentes funcionários. A mesma comissão pode ser paga para diferentes empresas e diferentes funcionários. Mensalmente é feito o pagamento para todos os vendedores. Quando um vendedor faz a venda, ele reporta a ordem para o sistema. O sistema então confirma a ordem para o cliente e as ordens são entregues para os clientes semanalmente.

1) Avaliação da Qualidade do Texto

Afirmiação	Escolha 1 e somente 1 opção
1. O texto apresentado descreve claramente o domínio/negócio	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
2. O texto apresentado descreve um domínio/negócio de fácil entendimento	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
3. O texto apresentado descreve informações úteis para o Levantamento de Requisitos	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente

2) Avaliação da cobertura dos Requisitos Funcionais em um Sistema de Informação

A lista abaixo foi gerada automaticamente pelo método usando apenas as informações do texto. Marque com um **X** os requisitos que você considera que **NÃO** seria um requisito funcional em um Sistema de Informação (SI).

ID	Descrição do RFD	NÃO é um Requisito Funcional em um SI
RF1	O sistema deve gerenciar Pagamento	
RF1.1	O sistema deve permitir atribuir Vendedor a Pagamento	
RF1.1.1	Vendedor é um/uma Funcionário	
RF1.1.1.1	Funcionário é um/uma Pessoa	
RF1.1.1.1.1	O sistema deve manter os dados de Pessoa	
RF1.1.2	Vendedor é um/uma Empresa	
RF1.1.2.1	O sistema deve manter os dados de Empresa	
RF1.2	O sistema deve permitir atribuir Empresa a Pagamento	
RF2	O sistema deve gerenciar Venda	
RF2.1	O sistema deve permitir atribuir Cliente a Venda	
RF2.1.1	Cliente é um/uma Pessoa	
RF2.2	O sistema deve permitir atribuir Vendedor a Venda	
RF2.3	O sistema deve permitir gerar Ordem de Venda	
RF2.3.1	O sistema deve manter os dados de Ordem de Venda	

1. Você considera que o quadro acima representa <u>quantos por cento</u> do total de requisitos que é possível extrair por meio da leitura do texto?	(1) 0% (2) 25% (3) 50% (4) 75% (5) 100%
--	---

Texto 2 – Biblioteca

Uma biblioteca empresta itens para clientes. Cada cliente é conhecido como um membro o qual possui um cartão de membro que apresenta um número único de membro. Além do número do membro, outros detalhes do cliente devem ser mantidos tais como, nome, endereço e data de nascimento. A biblioteca é composta de números de seções temáticas. Cada seção é identificada por uma marca de classificação. Um item de empréstimo é identificado unicamente por um código de barras. Há dois tipos de itens de empréstimo, fitas de idioma e livros. Uma fita de idioma tem o título e o nível. Um livro tem um título e o (s) autor (es). Um cliente pode emprestar no máximo 8 itens. Um item pode ser emprestado, reservado ou renovado. Quando um item é emprestado o cartão de membro do cliente é lido por meio de um leitor de código de barras ou é informado manualmente. Se o membro ainda é válido e o número de itens emprestados é menor que 8, o código de barras do livro é lido, também via leitor de código de barras ou informado manualmente. Se o item pode ser emprestado (ou seja, não está reservado) o item é carimbado e então emprestado. A livraria deve facilitar a busca dos itens e manter atualizações dos empréstimos diariamente.

1) Avaliação da Qualidade do Texto

Afirmiação	Escolha 1 e somente 1 opção
1. O texto apresentado descreve claramente o domínio/negócio	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
2. O texto apresentado descreve um domínio/negócio de fácil entendimento	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
3. O texto apresentado descreve informações úteis para o Levantamento de Requisitos	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente

2) Avaliação dos Requisitos Funcionais em um Sistema de Informação

A lista abaixo foi gerada automaticamente pelo método usando apenas as informações do texto. Marque com um **X** os requisitos que você considera que **NÃO** seria um requisito funcional em um Sistema de Informação (SI).

ID	Descrição	NÃO é um Requisito Funcional em um SI
RF1	O sistema deve gerenciar Empréstimo	
RF1.1	O sistema deve permitir atribuir Cliente a Empréstimo	
RF1.1.1	Cliente é um/uma Pessoa	
RF1.1.1.1	O sistema deve manter dados de Pessoa	
RF1.1.2	O sistema deve permitir associar Cartão do Membro a Cliente	
RF1.1.2.1	O sistema deve manter dados de Cartão do Membro	
RF1.2	O sistema deve permitir atribuir Item a Empréstimo	
RF1.2.1	O sistema deve manter dados de Item	
RF1.2.1.1	O sistema deve manter dados de Livro	
RF1.2.1.1.1	O sistema deve permitir associar Autor a Livro	
RF1.2.1.1.1.1	Autor é um/uma Pessoa	
RF1.2.1.2	O sistema deve manter dados de Fita de Idioma	
RF1.2.2	O sistema deve permitir associar Seção a Item	
RF1.2.2.1	O sistema deve manter dados de Seção	
RF1.2.2.1.1	O sistema deve permitir associar Biblioteca a Seção	
RF1.2.2.1.1.1	O sistema deve manter dados de Biblioteca	

1. Você considera que o quadro acima representa <u>quantos por cento</u> do total de requisitos que é possível extrair por meio da leitura do texto?	(1) 0% (2) 25% (3) 50% (4) 75% (5) 100%
--	---

FINALIZAÇÃO DO EXPERIMENTO

Afirmção	Escolha 1 e somente 1 opção
1. As listas geradas pelo método representaram um conjunto coerente de Requisitos Funcionais candidatos em um Sistema de Informação	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
2. As listas geradas pelo método apresentadas de forma hierárquica podem ser úteis para apoiar a extração dos Requisitos Funcionais candidatos em um Sistema de Informação	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
3. Eu utilizaria um método que gerasse a lista de Requisitos Funcionais para apoiar a identificação dos Requisitos Funcionais do Software	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
4. Quais são as maiores dificuldades que você observa no seu dia-a-dia com relação ao Levantamento de Requisitos de Software?	
5. Descreva observações positivas e/ou negativas relacionadas ao experimento	
6. Hora fim do experimento	

INSTRUMENTO 2**EXPERIMENTO PARA AVALIAR A COBERTURA E UTILIDADE DE UM MÉTODO PARA EXTRAÇÃO DE REQUISITOS FUNCIONAIS DE DOMÍNIO****INFORMAÇÃO DO PARTICIPANTE**

Pergunta	Resposta do Participante
1. Hora início do experimento	
2. Nome	
3. Cargo/função profissional	
4. Curso de formação na graduação	
5. Nível de formação na área	Graduação <input type="checkbox"/> Especialização <input type="checkbox"/> Mestrado <input type="checkbox"/> Doutorado <input type="checkbox"/>
6. Já atuou/atua recebendo os requisitos prontos para o desenvolvimento do software	Sim <input type="checkbox"/> Não <input type="checkbox"/>
7. Se a resposta a questão 6 for Sim , por quantos anos	
8. Já atuou/atua diretamente com o Levantamento (Descoberta) de Requisitos de Software	Sim <input type="checkbox"/> Não <input type="checkbox"/>
9. Se a resposta a questão 8 for Sim , por quantos anos	
10. Com relação ao Levantamento de Requisitos de Software, qual é o seu grau de experiência?	(1) Sem experiência (2) Pouco experiente (3) Razoavelmente experiente (4) Experiente (5) Muito experiente

EXPLICAÇÃO DO EXPERIMENTO

- Este experimento trata da avaliação de um método que extrai Requisitos Funcionais a partir de textos.
- Os Requisitos Funcionais são extraídos em alto nível e são apresentados como possíveis candidatos a um requisito funcional em um Sistema de Informação.
- O método lista os Requisitos Funcionais de forma hierárquica considerando as dependências e as relações entre os conceitos identificados pelo método.
- A seguir são apresentadas as descrições de alguns domínios/negócios, a partir dos quais você poderia extrair uma lista inicial dos possíveis Requisitos Funcionais de um Sistema de Informação. Considerando apenas o que está descrito você deverá:
 - 1) Avaliar a qualidade do texto (respondendo 3 questões)
 - 2) Avaliar a cobertura dos Requisitos Funcionais gerados pelo método

EXECUÇÃO DO EXPERIMENTO

Texto 1 – Editora

Uma publicação está relacionada a livros. Editores publicam livros. Autores escrevem livros. Um livro pode ser escrito por vários autores. Um autor tem um nome, data de aniversário e endereço. Um livro tem um título e um preço. Um número ISBN identifica um livro. Cada livro é publicado exatamente por um editor. Um editor tem uma denominação e um endereço. Um editor tem funcionários. Os funcionários têm um número de seguro único, um nome e data de aniversário.

1) Avaliação da Qualidade do Texto

Afirmção	Escolha 1 e somente 1 opção
1. O texto apresentado descreve claramente o domínio/negócio	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
2. O texto apresentado descreve um domínio/negócio de fácil entendimento	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
3. O texto apresentado descreve informações úteis para o Levantamento de Requisitos	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente

2) Avaliação da cobertura dos Requisitos Funcionais em um Sistema de Informação

A lista abaixo foi gerada automaticamente pelo método usando apenas as informações do texto. Marque com um **X** os requisitos que você considera que **NÃO** seria um requisito funcional em um Sistema de Informação.

ID	Descrição	NÃO é um Requisito Funcional em um SI
RF1	O sistema deve gerenciar Publicação	
RF1.1	O sistema deve permitir atribuir Livro a Publicação	
RF1.1.1	O sistema deve manter os dados de Livro	
RF1.2	O sistema deve permitir atribuir Autor a Publicação	
RF1.2.1	Autor é um/uma Pessoa	
RF1.2.1.1	O sistema deve manter os dados de Pessoa	
RF1.3	O sistema deve permitir atribuir Editor a Publicação	
RF1.3.1	Editor é um/uma Pessoa	
RF1.3.2	O sistema deve permitir associar Funcionário a Editor	
RF1.3.2.1	Funcionário é um/uma Pessoa	

1. Você considera que o quadro acima representa <u>quantos por cento</u> do total de requisitos que é possível extrair por meio da leitura do texto?	(1) 0% (2) 25% (3) 50% (4) 75% (5) 100%
--	---

Texto 2 – Conferência

Um comitê organizador é composto por membros e este comitê é responsável por organizar uma conferência. O membro presidente do comitê organizador é o Chair, quem publica a conferência. Autores realizam a submissão de artigos, os quais são recebidos pelos membros do comitê. O comitê distribui os artigos entre os revisores, os quais são responsáveis pela condução da revisão e recomendação de quais artigos devem ser aceitos. Um artigo pode ter três estágios diferentes durante este ciclo, antes das revisões o artigo é definido como “artigo não avaliado”, depois das revisões, os artigos são definidos como “artigo aceito” ou “artigo recusado”. As apresentações são preparadas pelos autores baseadas somente nos artigos aceitos.

1) Avaliação da Qualidade do Texto

Afirmção	Escolha 1 e somente 1 opção
1. O texto apresentado descreve claramente o domínio/negócio	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
2. O texto apresentado descreve um domínio/negócio de fácil entendimento	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
3. O texto apresentado descreve informações úteis para o Levantamento de Requisitos	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente

2) Avaliação dos Requisitos Funcionais em um Sistema de Informação

A lista abaixo foi gerada automaticamente pelo método usando apenas as informações do texto. Marque com um **X** os requisitos que você considera que **NÃO** seria um requisito funcional em um Sistema de Informação.

ID	Descrição	NÃO é um Requisito Funcional em um SI
RF1	O sistema deve gerenciar Revisão	
RF1.1	O sistema deve permitir atribuir Revisor a Revisão	
RF1.1.1	Revisor é um/uma Pessoa	
RF1.1.1.1	O sistema deve manter dados de Pessoa	
RF1.2	O sistema deve permitir atribuir Artigo a Revisão	
RF1.2.1	O sistema deve manter dados de Artigo	
RF1.2.2.1	O sistema deve permitir informar Artigo Rejeitado a Artigo	
RF1.2.3.2	O sistema deve permitir informar Artigo Não Avaliado a Artigo	
RF1.2.3.3	O sistema deve permitir informar Artigo Aceito a Artigo	
RF2	O sistema deve gerenciar Submissão	
RF2.1	O sistema deve permitir atribuir Autor a Submissão	
RF2.1.1	Autor é um/uma Pessoa	
RF2.2	O sistema deve permitir atribuir Artigo a Submissão	
RF3	O sistema deve gerenciar Apresentação	
RF3.1	O sistema deve permitir atribuir Artigo Aceito a Apresentação	
RF4	O sistema deve gerenciar Organização Conferência	
RF4.1	O sistema deve permitir atribuir Comitê Organizador a Organização Conferência	
RF4.1.1	Comitê Organizador é um/uma Pessoa	
RF4.2	O sistema deve permitir atribuir Chair a Organização Conferência	
RF4.2.1	Chair é um/uma Comitê Organizador	
RF4.3	O sistema deve permitir atribuir Conferência a Organização Conferência	
RF4.3.1	O sistema deve manter dados de Conferência	

1. Você considera que o quadro acima representa quantos por cento do total de requisitos que é possível extrair por meio da leitura do texto?	(1) 0% (2) 25% (3) 50% (4) 75% (5) 100%
---	---

FINALIZAÇÃO DO EXPERIMENTO

Afirmção	Escolha 1 e somente 1 opção
1. As listas geradas pelo método representaram um conjunto coerente de Requisitos Funcionais candidatos em um Sistema de Informação	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
2. As listas geradas pelo método apresentadas de forma hierárquica podem ser úteis para apoiar a extração dos Requisitos Funcionais candidatos em um Sistema de Informação	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
3. Eu utilizaria um método que gerasse a lista de Requisitos Funcionais para apoiar a identificação dos Requisitos Funcionais do Software	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
4. Quais são as maiores dificuldades que você observa no seu dia-a-dia com relação ao Levantamento de Requisitos de Software?	
5. Descreva observações positivas e/ou negativas relacionadas ao experimento	
6. Hora fim do experimento	

INSTRUMENTO 3**EXPERIMENTO PARA AVALIAR A COBERTURA E UTILIDADE DE UM MÉTODO PARA EXTRAÇÃO DE REQUISITOS FUNCIONAIS DE DOMÍNIO****INFORMAÇÃO DO PARTICIPANTE**

Pergunta	Resposta do Participante
1. Hora início do experimento	
2. Nome	
3. Cargo/função profissional	
4. Curso de formação na graduação	
5. Nível de formação na área	Graduação <input type="checkbox"/> Especialização <input type="checkbox"/> Mestrado <input type="checkbox"/> Doutorado <input type="checkbox"/>
6. Já atuou/atua recebendo os requisitos prontos para o desenvolvimento do software	Sim <input type="checkbox"/> Não <input type="checkbox"/>
7. Se a resposta a questão 6 for Sim , por quantos anos	
8. Já atuou/atua diretamente com o Levantamento (Descoberta) de Requisitos de Software	Sim <input type="checkbox"/> Não <input type="checkbox"/>
9. Se a resposta a questão 8 for Sim , por quantos anos	
10. Com relação ao Levantamento de Requisitos de Software, qual é o seu grau de experiência?	(1) Sem experiência (2) Pouco experiente (3) Razoavelmente experiente (4) Experiente (5) Muito experiente

EXPLICAÇÃO DO EXPERIMENTO

- Este experimento trata da avaliação de um método que extrai Requisitos Funcionais a partir de textos.
- Os Requisitos Funcionais são extraídos em alto nível e são apresentados como possíveis candidatos a um requisito funcional em um Sistema de Informação.
- O método lista os Requisitos Funcionais de forma hierárquica considerando as dependências e as relações entre os conceitos identificados pelo método.
- A seguir é apresentada a descrição de um domínio/negócio, a partir do qual você poderia extrair uma lista inicial dos possíveis Requisitos Funcionais de um Sistema de Informação. Considerando apenas o que está descrito você deverá:
 - 1) Avaliar a qualidade do texto (respondendo 3 questões)
 - 2) Avaliar a cobertura dos Requisitos Funcionais gerados pelo método

EXECUÇÃO DO EXPERIMENTO

Texto 1 – Roteiro de Ônibus

Há duas maneiras de as pessoas viajarem com a ViagemMais. Os passageiros podem fazer a reserva da viagem ou os passageiros podem aparecer no portão de embarque sem uma reserva e comprar o bilhete de um assento livre. Para o passageiro com uma reserva é atribuída uma data de reserva, enquanto que para o passageiro sem reserva é atribuída a data de embarque. O nome e o endereço de todos os passageiros são coletados. Números de telefone são coletados quando possível.

Todas as viagens de ônibus são organizadas em segmentos de rotas diárias. Todos os segmentos de rotas diárias têm hora início e hora fim. Cada segmento de rota diária é classificado como um segmento de rota com o número do segmento, cidade início e cidade fim. A ViagemMais oferece um conjunto de viagens, cada viagem é composta por um ou mais segmentos de rota. Para cada viagem, há o número da viagem, cidade início e cidade fim. Se a viagem é organizada para um evento especial, o nome do evento também é associado. Cada segmento de rota diária que a ViagemMais oferece é parte de uma viagem diária. A viagem diária é realizada por um ou mais motoristas. O nome, endereço e número de todos os motoristas são coletados. A ViagemMais registra os motoristas que não compareceram. Quando um motorista falta, a ViagemMais registra a data de início e os detalhes da falta. Nesta situação o motorista deve informar as razões por ter faltado.

A ViagemMais também coleta informações sobre os ônibus utilizados nas viagens diárias. Os ônibus têm marca, modelo e um número de registro. Para os ônibus em uso, é coletado a média diária de quilometragem. Se o ônibus precisar de manutenção, a ViagemMais anota a data em que o ônibus entrou em manutenção e registra os problemas do ônibus. A ViagemMais atribui um número do problema e uma breve descrição para cada problema de manutenção. Finalmente, o custo de reparação de todos os problemas com o ônibus, também são registrados.

1) Avaliação da Qualidade do Texto

Afirmção	Escolha 1 e somente 1 opção
1. O texto apresentado descreve claramente o domínio/negócio	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
2. O texto apresentado descreve um domínio/negócio de fácil entendimento	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
3. O texto apresentado descreve informações úteis para o Levantamento de Requisitos	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente

2) Avaliação da cobertura dos Requisitos Funcionais em um Sistema de Informação

A lista abaixo foi gerada automaticamente pelo método usando apenas as informações do texto. Marque com um **X** os requisitos que você considera que **NÃO** seria um requisito funcional em um Sistema de Informação (SI).

ID	Descrição	NÃO é um Requisito Funcional em um SI
RF1	O sistema deve gerenciar Falta	
RF1.1	O sistema deve permitir atribuir Razão a Falta	
RF1.2	O sistema deve permitir atribuir Motorista a Falta	
RF1.2.1	Motorista é um/uma Pessoa	
RF1.2.1.1	O sistema deve manter dados de Pessoa	
RF2	O sistema deve gerenciar Segmento de Rota Diária	
RF2.1	O sistema deve permitir atribuir Segmento de Rota a Segmento de Rota Diária	
RF2.1.1	O sistema deve manter dados de Segmento de Rota	
RF2.1.1.1	O sistema deve permitir associar Cidade a Segmento de Rota	
RF2.1.1.1.1	O sistema deve manter os dados de Cidade	

RF3	O sistema deve gerenciar Viagem Diária	
RF3.1	O sistema deve permitir atribuir Ônibus a Viagem Diária	
RF3.1.1	O sistema deve manter dados de Ônibus	
RF3.1.1.1	O sistema deve permitir informar Marca a Ônibus	
RF3.1.1.2	O sistema deve permitir informar Modelo a Ônibus	
RF3.2	O sistema deve permitir atribuir Viagem a Viagem Diária	
RF3.3	O sistema deve permitir atribuir Motorista a Viagem Diária	
RF3.4	O sistema deve permitir atribuir Passageiro a Viagem Diária	
RF3.4.1	Passageiro é um/uma Pessoa	
RF4	O sistema deve gerenciar Reserva	
RF4.1	O sistema deve permitir atribuir Segmento de Rota Diária a Reserva	
RF4.2	O sistema deve permitir atribuir Passageiro a Reserva	
RF5	O sistema deve gerenciar Viagem	
RF5.1	O sistema deve permitir atribuir Segmento de Rota a Viagem	
RF5.2	O sistema deve permitir atribuir Cidade a Viagem	
RF6	O Sistema deve gerenciar Manutenção	
RF6.1	O sistema deve permitir atribuir Problema de Manutenção a Manutenção	
RF6.2	O sistema deve permitir atribuir Ônibus a Manutenção	

1. Você considera que o quadro acima representa <u>quantos por cento</u> do total de requisitos que é possível extrair por meio da leitura do texto?	(1) 0% (2) 25% (3) 50% (4) 75% (5) 100%
--	---

FINALIZAÇÃO DO EXPERIMENTO

Afirmiação	Escolha 1 e somente 1 opção
1. A lista representou um conjunto coerente de Requisitos Funcionais candidatos em um Sistema de Informação	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
2. A lista de Requisitos Funcionais apresentada de forma hierárquica pode ser útil para apoiar a extração dos Requisitos Funcionais candidatos em um Sistema de Informação	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
3. Eu utilizaria um método que gerasse a lista de Requisitos Funcionais para apoiar a identificação dos Requisitos Funcionais do Software	(1) Discordo totalmente (2) Discordo (3) Não concordo nem discordo (4) Concordo (5) Concordo totalmente
4. Quais são as maiores dificuldades que você observa no seu dia-a-dia com relação ao Levantamento de Requisitos de Software?	
5. Descreva observações positivas e/ou negativas relacionadas ao experimento	
6. Hora fim do experimento	

APÊNDICE D – TEXTOS UTILIZADOS NO EXPERIMENTO PARA AVALIAR A IDENTIFICAÇÃO AUTOMÁTICA DOS CONSTRUTOS ONTOUML

Texto 1.1

Vendors may be employees or companies. Employees receive a salary amount and a commission amount, whereas companies only receive a commission amount. Each order corresponds to one vendor only, and each vendor has made at least one order, which is identified by an order number. The same salary may be paid to different sales employees. The same commission amount may be paid to different companies and to different sales employees. A monthly payment is made to all vendors. When a vendor makes a sale, he/she reports the order to the system. The system then confirms the order to the customer, and orders are delivered to customers weekly.

Texto 1.2

A library issues loan items to customers. Each customer is known as a member and is issued a membership card that shows a unique member number. Along with the membership number, other details on a customer must be kept such as a name, address, and date of birth. The library is made up of a number of subject sections. Each section is denoted by a classification mark. A loan item is uniquely identified by a bar code. There are two types of loan items, language tapes, and books. A language tape has a title language (e.g., French), and level (e.g., beginner). A book has a title, and author(s). A customer may borrow up to a maximum of 8 items. An item can be borrowed, reserved or renewed to extend a current loan. When an item is issued the customer's membership number is scanned via a bar code reader or entered manually. If the membership is still valid and the number of items on loan less than 8, the book bar code is read, either via the bar code reader or entered manually. If the item can be issued (e.g., not reserved) the item is stamped and then issued. The library must support the facility for an item to be searched and for a daily update of records.

Texto 1.3

Candidate will register with system to hire services. Candidate can provide information about his academic, work experience and referee details. Candidates may make verification request during registration or some time later. Candidate updates his details at any time. System will inform service seeker about approximate time period required to provide service. System records details of candidate. System shall keep status of each request up-to-date. System shall interact with education institute systems to verify originality of degree. System asks type of service required. Service may be Standard, Silver or Gold. System enters request as a record in system. System informs about outcome to the candidate. Referees send recommendation letters using system on behalf of students. Customers pay fee for each type of service. Customer may be candidate or employer. Standard service verifies details of education. Silver service verifies details of education and profession. Gold service verifies details of education, profession and recommendation letters from referees. Employer accesses system to hire services of system for verification purposes. Employer registers with system. After, employer should provide system with information on type of service required along with candidate unique identification number. System records details of employer. Verification officer accesses system to retrieve verification and inquiry requests. Verification officer performs all types of verifications requested by candidate and employers upon retrieving requests. Verification officer also verifies authenticity of referee. Verification officer uses system to send to referees a request for a recommendation letter when candidate requests gold service.

Texto 2.1

Publishers publish books. Authors write books. A book can be written by several authors. An author has a unique name, a birth date and an address. A book has a unique title and a price. An ISBN number identifies a book. Each book is published by exactly one publisher. A publisher has a unique denomination and an address. A publisher has employees. An employee has a unique social insurance number, a name and a birth date.

Texto 2.2

An Organizing Committee is composed by Committee Members and is responsible of organizing the Conference. The presiding Member of the Organizing Committee is the Chair, who publishes the Conference. Authors perform the submission of Papers, which are received by Committee Members who distribute them among Reviewers who are responsible for conducting reviews and recommending which papers should be accepted. A Paper can have three different stages during its lifecycle, before the reviews it is marked as a "Not Evaluated Paper", after the review, it becomes either an "Accepted Paper" or a "Rejected Paper". Presentations are prepared by Authors based on Accepted Papers only.

Texto 2.3

There are two ways for people to travel with Voyager. Either passengers can make a reservation on a trip, or passengers can show up at the boarding gate without a reservation and purchase a ticket for an unreserved seat. Passengers with a reservation are assigned a reservation date, whereas, passengers without reservations are assigned a boarding date. The name and addresses of all passengers are collected. Telephone numbers are collected where possible. All bus trips are organized into daily route segments. All daily route segments have both a start time and an end time. Each daily route segment. Voyager organizes is classified as a route segment with a segment number, start town, and finish town. Voyager offers a range of trips, and each trip is made up of one or more route segments. For every trip there is a trip number, start town, and finish town. If the trip is organized around a special event, the event name is also associated with the trip. Each daily route segment that Voyager offers is part of a dally trip. A daily trip is undertaken by one or more bus drivers. The name, address, and employee number of all drivers is collected. Voyager also records information about absent drivers. When a driver is absent. Voyager records the absence start date and the details about the absence. The absent driver provides one or more reasons for being absent and each reason is assigned a detail number and a short description. Voyager also collects information about the buses used for daily trips. Buses have a make, model, and registration number. For buses in use, the average daily kilometers is collected. If a bus requires maintenance, Voyager notes the date on which the bus entered maintenance and records the one or more problems with the bus. Voyager assigns a problem number and a short description for every maintenance problem. Finally, the average cost to repair all problems with a bus in maintenance is also recorded.

APÊNDICE E – RESULTADOS BRUTOS DA AVALIAÇÃO DA IDENTIFICAÇÃO AUTOMÁTICA DOS CONSTRUCTOS ONTOUML

Texto	Termos	Problema desambiguação	Sentido	Tipo semântico	Especialista					Consenso	Construto	Especialista					Consenso	
					1	2	3	4	5			1	2	3	4	5		
1.1	commission amount		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Sim	Sim	Sim	datatype
1.1	company		group	social group	Sim	Sim	Sim	Sim	Sim	Sim	kind	Sim	Sim	Sim	Sim	Sim	Não collective	kind
1.1	customer		person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Sim	Sim	Sim	Sim	Sim	role
1.1	employee		person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Sim	Sim	Sim	Sim	Sim	role
1.1	order	x	communicatio	artefato	Sim	Não	Sim	Sim	Sim	Sim	kind	Sim	-	Sim	Sim	Sim	Não action	kind
1.1	order number		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Sim	Sim	Sim	datatype
1.1	payment	x	act	activity	Sim	Sim	Sim	Sim	Sim	Sim	relator	Sim	Não quality	Sim	Sim	Sim	Não action	relator
1.1	sale		act	activity	Sim	Sim	Sim	Sim	Sim	Sim	relator	Sim	Sim	Sim	Sim	Sim	Não action	relator
1.1	sale employee		-	regra ii	Sim	Não	Sim	Sim	Sim	Sim	association	Sim	-	Não	Sim	Sim	Não role	sem consenso
1.1	vendor		person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Role	Sim	Role	Mixi	Sim	role
1.1	salary		possession	cost	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	quar	Sim	quality	Não kind	Sim	Não kind
1.1	salary amount		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Sim	Sim	Sim	datatype
1.2	author		person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Não datatype	Sim	Não datatype	Sim	Sim	role
1.2	book		communicatio	artefato	Sim	Sim	Sim	Sim	Sim	Sim	kind	Sim	Não subkind	Sim	Não subkind	Sim	Sim	kind
1.2	customer		person	rank	Sim	Sim	Sim	Não	Sim	Sim	role	Sim	Sim	Sim	-	Sim	Sim	role
1.2	item	x	artifact	artefato	Sim	Não	Sim	Sim	Sim	Sim	kind	Sim	Sim	Sim	Sim	Sim	Sim	kind
1.2	loan item		-	regra ii	Sim	Sim	Não	Sim	Sim	Sim	association	Não role	Não kind	-	Sim	Não category	sem consenso	
1.2	library	x	artifact	artefato	Sim	Não	Não	Sim	Sim	Sim	kind	Sim	-	-	Sim	Sim	kind	
1.2	loan		possession	other	Sim	Sim	Sim	Sim	Sim	Sim	relator	Sim	Sim	Sim	Sim	Sim	relator	
1.2	member		person	rank	Sim	Não	Não	Sim	Sim	Sim	role	Sim	-	-	Sim	Sim	role	
1.2	membership card		communicatio	artefato	Sim	Sim	Sim	Sim	Sim	Sim	kind	Sim	Sim	Sim	Sim	Sim	kind	
1.2	section	x	ognition	other	Não	Sim	Não	Sim	Sim	Sim	mode	-	Não kind	-	Sim	Não collective	sem consenso	
1.2	language tape		-	-	Sim	Sim	Sim	Sim	Sim	Sim	-	Não kind	Não subkind	Não kind	Não subkind	Não kind	falha método	
1.2	bar code		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Sim	Sim	datatype	
1.2	book bar code		-	regra i	Não	Não	Não	Sim	Sim	Não	datatype	-	-	-	Sim	Sim	não relevante	
1.2	classification mark		-	-	Não	Sim	Não	Sim	Sim	Sim	-	Não datatype	-	Não datatype	Não kind	Sim	sem consenso	
1.2	detail		ognition	other	Sim	Não	Sim	Não	Sim	Sim	datatype	Sim	-	Sim	-	Sim	datatype	
1.2	member number		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Sim	Sim	datatype	
1.2	membership number		-	regra i	Sim	Não	Não	Não	Sim	Não	datatype	Sim	-	-	-	Sim	não relevante	
1.2	number		attribute	states & propertie	Sim	Não	Não	Não	Não	Não	datatype	Sim	-	-	-	Sim	não relevante	
1.2	title	x	communicatio	language	Não	Sim	Sim	Sim	Não	Sim	datatype	-	Sim	Sim	Sim	Sim	datatype	
1.2	title language		-	-	Não	Sim	Não	Sim	Sim	Sim	-	-	Não datatype	-	Não datatype	Não datatype	falha método	
1.3	candidate		person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Sim	Sim	Sim	Sim	role	
1.3	customer		person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Sim	Sim	Não	Rolemixi	Sim	
1.3	employer		person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Sim	Sim	Sim	Sim	role	
1.3	education		ognition	other	Não	Não	Sim	Sim	Sim	Sim	mode	-	-	Sim	Sim	Não relator	sem consenso	
1.3	institute		group	social group	Sim	Sim	Sim	Sim	Sim	Sim	kind	Sim	Sim	Sim	Sim	Não	Não collective	
1.3	gold service		-	-	Sim	Sim	Sim	Sim	Sim	Sim	-	Não relat	Não phase	Não kind	Não datatype	Não sevice	sem consenso	
1.3	fee		possession	cost	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	quar	Sim	datatype	Sim	relator	
1.3	profession		act	activity	Sim	Não	Sim	Sim	Sim	Sim	relator	Não role	-	Não kind	Sim	Sim	sem consenso	
1.3	experience	x	ognition	other	Sim	Não	Sim	Sim	Sim	Sim	mode	Sim	Sim	Sim	Sim	Sim	mode	
1.3	recommendation letter		-	-	Sim	Sim	Sim	Sim	Sim	Sim	-	Não kind	Não kind	Não kind	Não relator	Não kind	falha método	
1.3	referee		person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Sim	Sim	Sim	Sim	role	
1.3	request		communicatio	activity	Sim	Sim	Sim	-	Sim	Sim	relator	Sim	Sim	Sim	Sim	?	Não UFO-5	
1.3	service	x	act	activity	Sim	Sim	Sim	Sim	Sim	Sim	relator	Sim	Não kind	Não kind	Não ?	Não UFO-5	sem consenso	
1.3	silver service		-	-	Sim	Sim	Sim	Sim	Sim	Sim	-	Não relat	Não phase	Não kind	Não datatype	Não UFO-5	sem consenso	
1.3	student		person	rank	Sim	Sim	Não	Sim	Sim	Sim	role	Sim	Sim	Sim	Sim	Sim	role	
1.3	type	x	ognition	other	Não	Não	Sim	Não	Sim	Não	mode	-	-	Não catego	-	Não powertype	não relevante	
1.3	verification officer		person	rank	Sim	Não	Sim	-	Sim	Sim	role	Sim	-	Sim	?	Sim	role	
1.3	authenticity		attribute	states & propertie	Sim	Sim	Sim	-	Sim	Sim	datatype	Sim	Não relator	Sim	?	Sim	datatype	
1.3	behalf		act	activity	Não	Não	Não	-	Sim	Não	relator	-	-	-	-	?	não relevante	
1.3	candidate requests gold service		-	-	Não	Não	Não	-	Sim	Não	-	-	-	-	-	?	não relevante	
1.3	detail		ognition	other	Sim	Não	Não	-	Sim	Não	datatype	Sim	-	-	-	Sim	não relevante	
1.3	identification number		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Sim	Sim	datatype	
1.3	time period		time	time	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	?	Sim	Sim	datatype	
1.3	verification	x	communicatio	activity	Sim	Sim	Sim	Sim	Sim	Sim	relator	Não mod	Sim	Sim	Não	Não mode	sem consenso	
1.3	verification purposes		-	-	Não	Não	Não	Não	Sim	Não	-	-	-	-	-	?	não relevante	

Figura 10-2. Avaliação dos termos pelos especialistas, instrumento 1

2.1	address	x	location	place	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Sim	Sim	datatype					
2.1	author		person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Sim	Sim	Sim	Não	role					
2.1	book		communicatio	artefato	Sim	Sim	Sim	Sim	Sim	Sim	kind	Sim	Sim	Sim	Sim	Sim	kind					
2.1	employee		person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Sim	Sim	Sim	Sim	role					
2.1	publisher		person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Não	kind/sub	Sim	Sim	role					
2.1	denomination	x	communicatio	language	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	quality	Sim	datatype					
2.1	title	x	communicatio	language	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Não	quality	Sim	datatype					
2.1	birth date		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Não	quality	Sim	datatype					
2.1	insurance number		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Não	quality	Sim	datatype					
2.1	isbn number		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Não	quality	Sim	datatype					
2.1	name		communicatio	language	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Não	quality	Sim	datatype					
2.1	price		attribute	states & propertie	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Não	quality	Sim	datatype					
2.2	accepted paper		-	-	Sim	Sim	Sim	Sim	Sim	Sim	-	-	role	-	phase	-	phase	falha método				
2.2	author		person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Sim	Sim	Sim	Sim	role					
2.2	chair	x	person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Sim	Sim	Sim	Sim	role					
2.2	conference		group	social group	Sim	Sim	Sim	Sim	Sim	Sim	kind	Sim	Sim	Sim	Não	event	kind					
2.2	committee members		-	-	Não	Sim	Não	Sim	Não	Não	-	-	Não	role	-	-	não relevante					
2.2	organizing committee		-	-	Sim	Sim	Sim	Sim	Sim	Sim	-	Não	colle	Não	collective	Não	collective	falha método				
2.2	paper		communicatio	artefato	Sim	Sim	Sim	Sim	Sim	Sim	kind	Sim	Sim	Sim	Sim	Sim	kind					
2.2	presentation		act	activity	Sim	Sim	Sim	Sim	Sim	Sim	relator	Não	kind	Sim	Sim	Não	kind	relator				
2.2	rejected paper		-	-	Sim	Sim	Sim	Sim	Sim	Sim	-	-	role	-	phase	-	phase	falha método				
2.2	reviewer		person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Sim	Sim	Sim	Sim	role					
2.2	review		act	activity	Sim	Sim	Sim	Sim	Sim	Sim	relator	Sim	Sim	Sim	Sim	Sim	relator					
2.2	submission		communicatio	activity	Sim	Sim	Sim	Sim	Sim	Sim	relator	Sim	Sim	Sim	Sim	Sim	relator					
2.2	member		person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Sim	Sim	Sim	Sim	role					
2.2	stage		time	time	Não	Não	Não	Sim	Não	Não	datatype	-	-	-	-	-	não relevante					
2.3	address	x	location	place	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Não	role	Não	?	Sim	Sim	datatype			
2.3	bus		artifact	artefato	Sim	Sim	Sim	Sim	Sim	Sim	kind	Sim	Sim	Sim	Sim	Sim	Sim	kind				
2.3	bus driver		person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Sim	Sim	Sim	Sim	role					
2.3	bus trips		-	regra ii	Não	Sim	Sim	Sim	Sim	Sim	association	-	Não	mediatic	Não	relator	Não	event	Sim	collective	sem consenso	
2.3	driver		person	rank	Não	Não	Sim	Sim	Sim	Sim	role	-	Sim	Sim	Sim	Sim	role					
2.3	finish town		-	-	Sim	Não	Sim	Sim	Sim	Sim	-	Não	role	-	-	-	-	-	-	-	-	sem consenso
2.3	maintenance		act	activity	Sim	Sim	Sim	Sim	Sim	Sim	relator	Sim	Sim	Sim	Sim	Sim	Não	phase	relator			
2.3	maintenance problem		-	regra iii	Sim	Sim	Sim	Não	Sim	Sim	mode	Não	kind	Sim	Sim	-	Não	kind	sem consenso			
2.3	make		cognition	other	Sim	Sim	Sim	Sim	Sim	Sim	mode	Não	kind	Não	kind	Não	kind	Não	datatype	Não	datatype	falha método
2.3	model		cognition	other	Sim	Sim	Sim	Sim	Sim	Sim	mode	Não	kind	Não	kind	Sim	Não	datatype	Não	datatype	sem consenso	
2.3	passenger		person	rank	Sim	Sim	Sim	Sim	Sim	Sim	role	Sim	Sim	Sim	Sim	Sim	Não	phase	role			
2.3	problem	x	state	states & propertie	Não	Não	Não	Sim	Sim	Não	mode	-	-	-	-	Não	kind	Não	kind	não relevante		
2.3	reservation		communicatio	activity	Sim	Sim	Sim	Sim	Sim	Sim	relator	Sim	Sim	Sim	Sim	Sim	Sim	relator				
2.3	route segment		-	-	Sim	Sim	Sim	Sim	Sim	Sim	-	Não	kind	Não	relator	Não	kind	Não	kind	falha método		
2.3	start town		-	-	Sim	Não	Sim	Sim	Sim	Sim	-	Não	role	-	-	-	-	-	-	-	-	sem consenso
2.3	trip		act	activity	Sim	Sim	Sim	Sim	Sim	Sim	relator	Sim	Sim	Sim	Sim	Não	event	Não	collective	relator		
2.3	absence		state	states & propertie	Sim	Sim	Sim	Sim	Sim	Sim	mode	Não	relat	Não	relator	Não	relator	Não	?	-	-	falha método
2.3	absence start date		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Não	quality	Sim	Sim	datatype			
2.3	cost		attribute	states & propertie	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Não	quality	Sim	Sim	datatype			
2.3	date		time	time	Sim	Sim	Não	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Sim	Sim	Sim	datatype				
2.3	description		communications	speech act	Sim	Não	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Não	quality	Sim	Sim	datatype			
2.3	detail		cognition	other	Sim	Não	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Não	quality	Sim	Sim	datatype			
2.3	employee number		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Não	quality	Sim	Sim	datatype			
2.3	end time		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Não	quality	Sim	Sim	datatype			
2.3	event name		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Não	quality	Sim	Não	kind	datatype		
2.3	gate		artifact	artefato	Não	Não	Não	Sim	Não	Não	kind	-	-	-	-	Sim	-	-	-	-	-	não relevante
2.3	information		communicatio	-	Não	Não	Não	Sim	Não	Não	-	-	-	-	-	Não	datatype	-	-	-	-	não relevante
2.3	kilometer		quantity	quantity	Sim	Sim	Não	Sim	Sim	Sim	datatype	Sim	Sim	Sim	-	Sim	Sim	Sim	datatype			
2.3	name		communicatio	language	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Não	quality	Sim	Sim	datatype			
2.3	number		quantity	quantity	Sim	Não	Sim	Sim	Sim	Sim	datatype	Sim	-	Sim	Não	quality	Sim	Sim	datatype			
2.3	problem number		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Não	quality	Sim	Sim	datatype			
2.3	range		object	-	Não	Não	Não	Não	Não	Não	-	-	-	-	-	-	-	-	-	-	-	não relevante
2.3	reason		cognition	other	Sim	Sim	Sim	Sim	Sim	Sim	mode	Não	kind	Sim	Sim	Não	datatype	Não	Datatype	mode		
2.3	registration number		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Não	quality	Sim	Sim	datatype			
2.3	reservation date		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Não	quality	Sim	Não	Quality	datatype		
2.3	segment number		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Não	quality	Sim	Sim	datatype			
2.3	start time		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Não	quality	Sim	Sim	datatype			
2.3	telephone number		-	regra i	Sim	Sim	Sim	Sim	Sim	Sim	datatype	Sim	Sim	Sim	Não	quality	Sim	Sim	datatype			

Figura 10-3. Avaliação dos termos pelos especialistas, instrumento 2

texto	Especialista									
	1		2		3		4		5	
1.1	employment	Não encontrado	Nenhuma sugestão		company vendor	Não encontrado	employer	Não encontrado	commission	quantity
1.1	supplier	Não encontrado			employee vendor	Não encontrado	derivação salary	Não encontrado	system	kind
1.1	employer	Não encontrado					derivação sale	Não encontrado		
1.1	supplier payment	Não encontrado					sale customer	Não encontrado		
1.1	employee payment	Não encontrado					salary employee	Não encontrado		
1.2							salary employer	Não encontrado		
1.2	loaned book	Não encontrado	name	datatype	item type	Não encontrado	loan member	Não encontrado	Nenhuma sugestão	
1.2	loaned language tape	Não encontrado	address	datatype	reserve	Não encontrado	membership	relator		
1.2	item status	Não encontrado	date of birth	datatype			member membership	Não encontrado		
1.2	loan item status	Não encontrado	level	datatype			library membership	mediation		
1.2			borrowed	phase			loan derivation	derivation		
1.2			reserved	phase			membership derivation	derivation		
1.2			renewed	phase						
1.3	Nenhuma sugestão		academic information	Não encontrado	standard Service	kind	Nenhuma sugestão		Nenhuma sugestão	
1.3			work experience	kind	outcome	kind				
1.3			recommendation	relator						
1.3			service seeker	role						
1.3			status	datatype						
1.3			standard	phase						
2.1	person	Não encontrado	person	Não encontrado	person	Não encontrado	Nenhuma sugestão		person	Não encontrado
2.1	authorship	Não encontrado	authorship	Não encontrado	authorship	Não encontrado			authorship	Não encontrado
2.1	employment	Não encontrado	employment	Não encontrado	employment	Não encontrado			employment	Não encontrado
2.1	publication	Não encontrado			publication	Não encontrado			publishing	Não encontrado
2.1	organization	Não encontrado			organization	Não encontrado			individual author	Não encontrado
2.1									author group	Não encontrado
2.2	person	Não encontrado	person	Não encontrado	person	Não encontrado	publish	Não encontrado	person	Não encontrado
2.2	chairmanship	Não encontrado	not evaluated paper	phase	conference publication	Não encontrado	not evaluated	phase	not evaluated paper	phase
2.2					suggestion	Não encontrado				
2.3			absence reason	Não encontrado	average daily kms	quality				
2.3			absence reason number	Não encontrado	average repair cost	quality				
2.3			absent driver	role			absent driver	phase	absent driver	phase
2.3			boarding	relator						
2.3	boarding date	datatype	boarding date	datatype	boarding date	quality	boarding date	datatype		
2.3	employment	Não encontrado	bus in use	Não encontrado	free seat	Não encontrado				
2.3	event	kind	bus not in use	Não encontrado						
2.3			event trip	Não encontrado						
2.3			maintenance problem description	Não encontrado						
2.3			maintenance problem	Não encontrado			people	kind		
2.3			person	Não encontrado	person	Não encontrado			person	Não encontrado
2.3	purchase	relator	segment finish town	Não encontrado	purchase	relator				
2.3	ticket	kind	segment start town	Não encontrado	reserved seat	role	ticket	kind		
2.3					seat	kind			seat	kind
2.3	town	kind	town	kind	town	kind			town	kind
2.3			town name	Não encontrado						
2.3			trip bus	Não encontrado						
2.3			trip finish town	Não encontrado						
2.3			trip number	datatype						
2.3			trip start town	Não encontrado						

Figura 10-4. Indicação de termos relevantes não identificados pelo método

APÊNDICE F – RESULTADOS BRUTOS DA AVALIAÇÃO DE DERIVAÇÃO DE REQUISITOS FUNCIONAIS DE DOMÍNIO

ID Participante	Maiores dificuldades	Observações experimento
1	Falta de informações consistentes Tempos nos cronogramas Falta de uso de técnicas adequadas Requisitos voláteis Requisitos incompletos	Textos muito diretos Mapeamento de cardinalidade ajudaria
2	Compreender o domínio do negócio Alinhar o vocabulário do domínio ao dos desenvolvedores	Lista de forma hierarquica ajuda a compreender o contexto dos requisitos
3	Entender de forma concisa e clara as reais necessidades dos clientes	Gostei bastante, deixa claro o entendimento dos requisitos
4	precisa	Gostei da forma como o método trata e representa os requisitos em lista
5	Identificar alguns requisitos implícitos que está no texto	Experimento bem elaborado
6	Compreensão do domínio e regras de negócio	Faltaram requisitos A organização hierarquica facilitou a análise dos requisitos levantados
7	Falta de clareza e detalhamento das respostas dos usuários Mudanças de requisitos	Negativo: na explicação colocar exemplos Positivo: texto bem elaborado A hierarquia facilita o entendimento do requisito
8	desconhecido ou quando o usuário tem dificuldades em descrever o que necessita	-
9	Acesso a especialistas com conhecimento Compreensão adequada do domínio Tempo necessário para fazer com qualidade	-
10	Extrair a totalidade da compreensão da regra de negócio	-
11	-	Gerou requisitos inválidos, porém em pequena quantidade. Ponto positivo, faz correlacionamento de requisitos pertinentes que não está explícito no texto
12	Extrair os requisitos de forma correta Falta de clareza da parte dos usuários Requisitos ambíguos	-
13	Comunicação com o cliente	-
14	-	-
15	-	-
16	Stakeholder errado Requisitos não necessários	-
17	Saber exatamente o que o cliente deseja, o cliente não tem conhecimento do desenvolvimento o software e tem dificuldade em explicar um requisito de maneira direta e consistente	-
18	Identificar com clareza e detalhamento o domínio (texto) por parte dos stakeholders. Identificar as reais necessidades que devem ser controladas pelo SI	-
19	Compreender o negócio, o que pode ser automatizado, o que integrar, que informação tem valor	-
20	A existência de vários sistemas isolados, que não se conversam, e que acabam tornando o processo confuso. O ideal é efetuar primeiramente um levantamento dos processos atuais, para depois fazer uma sugestão de melhoria dos processos atuais. Somente depois de propor uma nova solução é que se deve partir para o levantamento de requisitos. Resistência e muitas vezes desconhecimento dos processos por parte dos usuários que irão repassar o conhecimento.	Achei interessante a ideia da utilização de um método automatizado para a geração da lista de requisitos. Observei, no entanto, que algumas regras de negócio foram confundidas com requisitos e muitos requisitos não foram relacionados. Acho que nada vai substituir o bom senso, a interpretação e a experiência do profissional que vai efetuar o levantamento de requisitos, porém, é uma ferramenta que pode auxiliar no processo como um todo.

Figura 10-5. Observações dos profissionais parte 1

21	A identificação de detalhes de regras de negócios como foi apontado no texto 2. Esse tipo de informação se perde ou fica omitida o que impacta os desenvolvedores no momento da codificação.	Não tenho absoluta certeza se identificar o Cliente como Pessoa ou funcionário como Pessoa faria diferença no levantamento de requisitos funcionais. Essa informação talvez é importante no momento da análise de sistemas.
22	Atualmente não trabalho mais com o levantamento de Requisitos, mas percebo na equipe, que a maior dificuldade deles é em como extrair os requisitos funcionais, além de terem dificuldades de extrair os requisitos "escondidos", pois o usuário geralmente, em sua entrevista, fala muito dos requisitos evidentes, pois é o que o usuário geralmente conhece e sabe que o sistema esta executando e muitos não conseguem identificar estes requisitos	O experimento nos faz perceber o quanto pode ser falho um levantamento de requisitos, neste em específico identifiquei itens positivos, como uma visão clara dos requisitos funcionais listados em uma hierarquia de fácil entendimento, além de evidenciar alguns requisitos "escondidos" que muitas vezes não são evidenciados no momento correto.
23	Falta de patrocínio devido a mudanças no governo e consequentemente em toda cadeia de chefias Falta de visão do todo (integrações, sistemas futuros), resultando em sistemas incompletos que precisam ser refeitos em curto espaço de tempo.	Método interessante que pode ser utilizado no dia a dia. Tive um pouco de dificuldade para preencher no celular o questionário, algumas tabulações ficaram estranhas na tela e acabei levando muito tempo para responder.
24	Transmissão de conhecimento entre cliente e analistas Habilidade do cliente perante a tecnologia Conhecimento de negócio baixo do cliente Analistas que querem ser o cliente	Se tiver um texto bem fundamentado a geração dos requisitos funcionais será boa. Em vários momentos o texto gerado apresentou determinadas entidades pessoa. Reitero que em domínio de negócio ela continua sendo uma entidade
25	cliente; É muito frequente o fato do analista de requisitos "ter que correr atrás" das definições junto ao cliente para poder fazer a análise de requisitos; 2. Falta de tempo hábil para a análise efetiva dos requisitos (elicitação , análise , refinamento e desenho); 3. Falta de tempo para verificação e validação dos requisitos . Essas etapas, pelo menos na empresa em que trabalho, exigem o envolvimento de pessoas envolvidas em outros projetos, fora do time de desenvolvimento e, muitas vezes, em locais de trabalho diferentes o que compromete a qualidade destas Falta de definições ou baixa qualidade das definições repassadas pelo cliente; É muito frequente o fato do analista de requisitos "ter que correr atrás" das definições junto ao cliente para poder fazer a análise de requisitos; 2. Falta de tempo hábil para a análise efetiva dos requisitos (elicitação , análise , refinamento e desenho); 3. Falta de tempo para verificação e validação dos requisitos . Essas etapas, pelo menos na empresa em que trabalho exigem o envolvimento de pessoas de outros projetos, fora do time de desenvolvimento e, muitas vezes, em locais de trabalho diferentes o que compromete a qualidade destas etapas	É bom contribuir com estudos, principalmente naqueles que podem melhorar nosso trabalho
26	já definidas, o que nem sempre o que o cliente precisa de fato. A partir disso, definimos uma lista de requisitos que o cliente entenda que vá lhe atender	Ótimo experimento
27	Os usuários não sabem o que querem e ficam na " tentativa e erro"	-
28	O cliente é muito superficial ao fazer uma solicitação	O experimento foi bem interessante, extraiu uma boa qualidade. Acredito que possa ajudar o analista no levantamento de um sistema
29	Identificação de requisitos primários uma vez que investimos tempo em questões e discussões relacionadas aos secundários	Experimento simples e de fácil entendimento
30	Entendimento da real necessidade dos stakeholders	Fácil leitura e entendimento. Fiquei em dúvida sobre requisitos do tipo relatório se deveriam ter aparecido ou não
31	Entendimento do Problema do Cliente	-
32	Entender qual é de fato o problema do cliente. Particularidades de cada cliente	Método e experimento bem interessante, auxiliaria bastante no dia-a-dia
33	Obter as informações necessárias para construir um sistema que atenda a necessidade	As questões são bem organizadas e claras
34	O requisito já chega como proposta de solução, sem explicitar qual é dificuldade ou problema do usuário/cliente	-
35	-	-
36	Dificuldade em entender a real necessidade do cliente. As vezes o cliente quer algo que ele não precisa	Os textos estavam escritos em forma de tópicos. A extração como foi apresentada pareceu ser simples
37	Identificar requisitos implícitos não mencionados pelo stakeholder, por, para ele. Ser trivial	Método sistemático produz uma base para redigir a lista de requisitos
38	Cliente passar uma visão limitada ao processo atual de trabalho. Necessidade de modificar o processo do cliente antes de encomendar o sistema	A forma hierárquica expõe as necessidades de maneira organizada, que facilita o entendimento. Somente o texto pode não ser suficiente para expressar todas as funcionalidades
39	Tempo dos stakeholders, dificuldade de envolver e trabalhar com todos os interessados no projeto, dificuldades de comunicação entre a equipe técnica e cliente, formato da documentação (textual, extenso), falta de comprometimento com as definições	-
40	-	quadro que contém a lista dos requisitos, a compreensão se torna muito melhor, mais fácil
41	Normas/estatutos informam os procedimentos mas na prática as ações são outras	Textos gerando dúvidas, demonstrando como é importante detalhar os reais requisitos
42	Falta de um processo bem definido	-

Figura 10-6. Observações dos profissionais parte 2

ID Participante	Maiores dificuldades	Observações experimento
1	A falta de entendimento entre o analista e o cliente	-
2	Identificar o escopo	Reduz o escopo facilitando o entendimento do negócio
3	Requisitos que poderiam ter ficado mais claros em um primeiro momento	Uma etapa antes para o participante identificar os requisitos antes de vê-los
4	-	-
5	-	Método muito interessante
6	-	Achei bastante interessante, acho que com a utilização desse método é possível facilitar o trabalho. É bastante confiável
7	Saber o que é preciso e qual é o fluxo do meu sistema	Os textos não representam muitas informações
8	Elicitar 100% dos requisitos em pouco tempo	-
9	-	Alguns requisitos são para a mesma funcionalidade porém com palavras diferentes
10	Os clientes não são claros ao explicar o negócio	-
11	Abstrair o contextos, quebrar em sub-requisitos	-
12	Identificação dos requisitos do escopo que nem sempre é bem definido	-
13	Falta de clareza do problema ou do que o cliente realmente quer	O levantamento de requisitos de forma hierárquica parece bem interessante e parece funcionar
14	A falta de conhecimento do domínio	Auxilia na formulação dos requisitos funcionais
15	Falta de informação ou informação incoerente	Gostei bastante do experimento e achei muito útil a extração dos requisitos.
16	Descobrir o que o cliente quer visualizar	Acredito que se o aluno fizesse o levantamento dos requisitos antes, seria mais valioso para o estudo
17	Interpretar o que deve ser considerado requisito	Pode não ser 100%, mas com certeza auxiliar no levantamento dos requisitos, principalmente para os menos experientes
18	O entendimento do domínio e a indecisão/imprecisão na hora de dizer o que ele quer	Experimento bem claro e objetivo
19	Falta de um escopo bem definido. Falta de boa descrição do negócio	Requisitos genéricos
20	Extrair de maneira clara e assertiva os requisitos	-
21	Relacionamento com o cliente. Na vida real não chega um texto bonito	-
22	Diferenciar requisitos funcionais de não funcionais	-
23	Informações mal expressadas não respeitando grau de hierarquia de informações, dificultando o entendimento	Iniciativa interessante pois tornaria o trabalho mais ágil. Pode servir de um material de consulta
24	-	Extração automática pareceu interessante, porém muitos detalhes seriam interessantes na modelagem mas não na transformação de requisitos
25	Compreender as regras de negócio a serem informatizadas	Referência errada de Pessoa
26	Pouca informação do negócio	Muito útil se uma ferramenta como esse existisse no mercado
27	Entendimento do negócio	-
28	-	Bem estruturado
29	Abstração das necessidades dos usuários	Ajuda na elicitação do requisitos, porém um pouco vaga em certas situações
30	Abstração do negócio para o SI	O método apontou os principais RF's necessários
31	Organizar a informação coletada	Interessante a forma de escrita dos requisitos
32	Compreender as necessidades do cliente	-
33	Falta de informação dos processos atuais	-
34	Compreender as necessidades do cliente	-
35	-	-
36	Identificação da necessidade do cliente Balanceamento entre o que é necessário e supérfluo Nível de detalhamento necessário para satisfazer o cliente	O método gerou alguns requisitos óbvios
37	Levantamento de todos os RFs relacionados ao sistema	-
38	Extrair do cliente todas as necessidades para o SI	Bom detalhamento das relações e atribuições
39	Entendimento do negócio	Dividiria o manter pessoa, pelo manter motorista, manter passageiro, etc
40	Entendimento do negócio	-
41	Conhecer o domínio, entender o que o usuário necessita	No 1o texto não achei muito bom o uso da ferramenta, mas acredito que o texto não estava muito claro. Já no 2o texto a lista de requisitos ficou mais elaborada

Figura 10-7. Observações dos estudantes