

# Avaliação de Abordagem Centrada em Modelos Conceituais em OntoUML para Apoiar a Derivação de Requisitos Funcionais

Joselaine Valaski <sup>1</sup>[0000-0003-3319-1715], Sheila Reinehr <sup>1</sup>[0000-0001-9430-7713] e Andreia Malucelli <sup>1</sup>[0000-0002-0929-1874]

<sup>1</sup> Programa de Pós-Graduação em Informática (PPGIA)  
Pontifícia Universidade Católica do Paraná (PUCPR), Curitiba-PR, Brasil  
joselaine.valaski@pucpr.br, sheila.reinehr@pucpr.br,  
malu@ppgia.pucpr.br

**Abstract.** A elicitação 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. Embora existam diversas técnicas para a elicitação de requisitos, algumas delas ainda são informais e dependem da experiência dos recursos humanos envolvidos. Neste sentido, um modelo conceitual pode ser um instrumento importante para apoiar a comunicação entre os envolvidos e explicitar conhecimentos de um domínio. 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 pesquisa tem como objetivo avaliar uma abordagem centrada em modelos conceituais OntoUML para apoiar a derivação de requisitos funcionais a partir de descrições de domínio. Os resultados apontam que é possível identificar semiautomaticamente os construtos OntoUML, bem como derivar requisitos funcionais a partir de descrições de domínios.

**Palavras-chave:** Elicitação de Requisitos, Modelo Conceitual, OntoUML.

## 1 Introdução

A elicitação de requisitos é a primeira atividade no processo de Engenharia de Requisitos de Software (ERS), e os problemas relacionados a esta atividade são os que mais afetam o sucesso dos projetos de desenvolvimento de software [1]. A falta de habilidade para discutir problemas tem sido uma das maiores deficiências da teoria e da prática em software [2]. A aquisição do conhecimento de um determinado domínio é um processo caro [3] pois necessidades mal compreendidas podem afetar negativamente etapas posteriores do desenvolvimento do software. Nas fases iniciais da ERS, nas quais ocorrem as primeiras interações com os *stakeholders*, são comuns problemas, tais como: imprecisão e deficiência na coleta de informações sobre o sistema a ser desenvolvido [4]; falta de entendimento do negócio pelos engenheiros de

requisitos [5]; as comunicações entre os especialistas no negócio e da computação são deficientes e comprometem a qualidade das informações [6]; e a falta de consenso sobre o uso dos termos na organização pode levar a significados distintos [7].

Na tentativa de atender ou minimizar estas e outras necessidades decorrentes dos problemas relacionados à ERS, as ontologias têm sido um importante recurso. As possibilidades de aplicação das ontologias na ERS são diversas, tendo um papel importante nas linguagens de modelagem conceitual. Mylopoulos [8] define a modelagem conceitual como a atividade de descrever formalmente aspectos do mundo físico e social com o propósito de entendimento. Nesta abordagem trata-se da modelagem da realidade e não da modelagem do sistema computacional [9]. Henderson-Sellers et al. [10] reforçam a importância da notação da linguagem ao afirmar que bons modelos conceituais precisam de boas linguagens de modelagem. 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 [3] propôs a linguagem OntoUML. O principal objetivo da OntoUML é atingir o comprometimento ontológico falho em outras linguagens genéricas para modelagem conceitual. A OntoUML possibilita maior expressividade nos modelos conceituais, evitando a sobrecarga e a redundância [3]. 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, frequentemente, escolhas de modelagem são feitas de maneira ad hoc [3].

No contexto da ERS, 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.

Considerando os problemas expostos anteriormente uma abordagem centrada em modelos conceituais em OntoUML para apoiar a derivação de requisitos funcionais a partir de descrições de domínio foi proposta sendo necessário avaliá-la. Este artigo tem como objetivo apresentar a avaliação da abordagem proposta. As avaliações foram realizadas em duas etapas distintas. A primeira envolvendo 10 especialistas em OntoUML, um público bem significativo, considerando o universo de especialistas disponíveis nesta área. A segunda etapa envolveu 42 profissionais da área de desenvolvimento de software e 41 estudantes de graduação da área de computação. A Seção 2 descreve a abordagem proposta, a Seção 3 apresenta o método aplicado para avaliação da abordagem, a Seção 4 apresenta e discute os resultados e, por fim, a Seção 5 finaliza o artigo com as conclusões.

## **2 Derivação de Requisitos Funcionais**

A construção da abordagem para apoiar a derivação de requisitos funcionais a partir de descrições de domínio foi executada em quatro etapas principais descritas a seguir.

### **2.1 Mapeamento e Revisão Sistemática da Literatura**

Nesta fase foram realizados um mapeamento [11-12] e uma revisão sistemática de literatura [13-14]. O mapeamento sistemático foi realizado com o objetivo de obter um panorama da área de Engenharia de Requisitos. A base do mapeamento foram 258 artigos publicados no Workshop em Engenharia de Requisitos (WER). Neste mapeamento 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 deixou evidente que ainda são necessários esforços na solução de problemas relacionados a estes temas. Já com o objetivo de compreender melhor como as ontologias estavam sendo usadas no processo da ERS foi realizada uma revisão sistemática de literatura. A revisão teve como objetivo identificar lacunas de pesquisa. A revisão iniciou com uma base de 2.407 artigos e foi concluída com a análise detalhada de 60 artigos.

Os resultados da revisão sistemática mostraram trabalhos efetivos principalmente para apoiar as atividades de análise (51,66%), elicitação (23,33%) e especificação (18,33%). As contribuições mais citadas pelas ontologias foram identificar problemas em especificação e modelos, melhorar a comunicação e construir modelos mais completos. Além disso, a revisão apontou que as ontologias têm como principal objetivo prover o entendimento de um domínio. As propostas também mostram a importância em se utilizar técnicas e linguagens de modelagem ontologicamente fundamentadas. O uso de uma linguagem com falhas de expressividade pode comprometer o entendimento dos artefatos de requisitos em fases posteriores. 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. A partir desta revisão, concluiu-se que modelos conceituais ontologicamente bem fundamentados podem ser um instrumento importante a ser utilizado na atividade de elicitação de requisitos. Detalhes sobre as abordagens propostas, referente aos 60 trabalhos identificados para o uso de ontologias em atividades pertinentes à Engenharia de Requisitos estão publicados em [14].

### **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 pesquisa. Embora Guizzardi [3] tenha ilustrado em sua tese situações em que a OntoUML possibilita

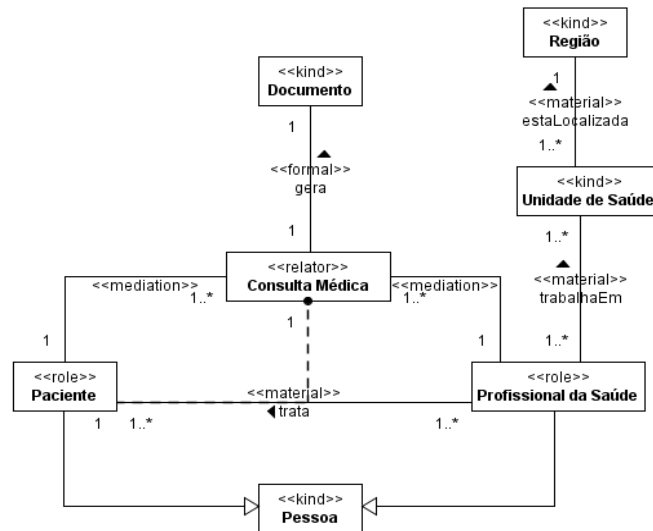
melhor representação evitando sobrecarga e redundância, considerou-se importante realizar uma avaliação dentro de um contexto de sistemas de informação para verificar a expressividade da OntoUML em relação a *Unified Modeling Language* (UML).

O primeiro experimento [15] foi realizado com um grupo de oito profissionais. Considerando doze sentenças pertinentes a um domínio e a dois modelos conceituais (um em UML e o outro em OntoUML), 13% indicaram a UML como a linguagem mais expressiva, 42% indicaram a OntoUML como a linguagem mais expressiva e 46% com ambas sendo expressivas no mesmo nível. O segundo experimento [15] foi realizado com um grupo de oitenta estudantes. Considerando as mesmas sentenças e os mesmos modelos apresentados aos profissionais, 19% indicaram a UML como a linguagem mais expressiva, 42% indicaram a OntoUML como a linguagem mais expressiva e 39% com ambas sendo expressivas no mesmo nível. 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 construir 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 [15]. Este resultado reforçou a motivação desta pesquisa, referente ao uso de modelos conceituais representados em OntoUML como instrumento para apoiar o processo de elicitação de requisitos.

### 2.3 Método para Derivação de Requisitos Funcionais

A avaliação da expressividade demonstrou diversas situações onde a OntoUML permite representações mais claras relacionados ao domínio. Este resultado reforçou a motivação desta pesquisa de que o modelo conceitual representado em OntoUML pode ser um instrumento relevante para apoiar a derivação de requisitos funcionais de domínio. Assim, um método formal para orientar a derivação de requisitos funcionais de domínio foi proposto [16]. Foram estudados nove modelos conceituais representados em OntoUML e, por meio deles, foi definida uma heurística com o objetivo de realizar a leitura e a transcrição textual destes modelos. O repositório de modelos conceituais em OntoUML, denominado *Menthor* (não está mais disponível na Web), foi utilizado como origem para a seleção de alguns modelos conceituais.

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 seis dos nove modelos inicialmente avaliados. Estes modelos foram analisados e padrões identificados. Baseados nestes padrões, uma heurística foi proposta. A Figura 1 apresenta um dos modelos conceituais selecionados e a Tabela 1 apresenta o resultado correspondente da heurística proposta.



**Figura 1.** Modelo conceitual parcial OntoUML (Conferência Acadêmica)

**Tabela 1.** Resultados parciais (Conferência Acadêmica)

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 ...	

Considerando o resultado dos nove modelos conceituais avaliados, a heurística proposta mostrou-se viável de ser utilizada. O método foi integrado a um ambiente computacional e posteriormente avaliado [16].

## 2.4 Ambiente para 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 [3] 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 atingir este objetivo foram integrados ao ambiente: algoritmo para a identificação de termos relevantes, algoritmo para desambiguação de termos, método para a identificação do construto OntoUML [17], ferramenta para a construção do modelo conceitual em OntoUML [18] e método para a derivação de requisitos de domínio. Por meio do ambiente é possível executar seis atividades principais, as quais são: 1) identificação de termos relevantes; 2) criação de arquivo XML com os termos relevantes taguados; 3) desambiguação e extração do *supersense* dos termos; 4) mapeamento entre os tipos semânticos e a OntoUML; 5) construção semiautomática do modelo; e 6) derivação da lista de requisitos funcionais.

Após a implementação do ambiente, um teste de viabilidade foi conduzido com o objetivo de avaliar o processo de construção semiautomática de um modelo conceitual em OntoUML. Para o teste foi selecionado um modelo conceitual, representado usando a notação Entidade-Relacionamento, e seu respectivo texto [19]. Este texto foi processado pelo ambiente, e os conceitos identificados foram comparados com os conceitos representados no modelo Entidade-Relacionamento. Os resultados do teste de viabilidade mostraram a integração e o funcionamento de diversos algoritmos e métodos que guiam o modelador na construção do modelo conceitual. Com relação ao desempenho geral do ambiente, no contexto avaliado, o ambiente conseguiu identificar automaticamente 60% dos conceituais significativos de um modelo conceitual [20-21]. Talvez seja um resultado inexpressivo para modeladores experientes, porém é útil para modeladores em fase de formação, os quais são o usuário-alvo do ambiente desenvolvido.

## 3 Método para Avaliação da Abordagem Proposta

Duas etapas foram executadas para avaliar a abordagem proposta, os detalhes de cada etapa são apresentados a seguir.

### 3.1 Avaliação da Identificação Automática dos Elementos

Na primeira etapa, seis descrições de domínio foram selecionadas a partir de publicações científicas que discutiam o uso de processamento de linguagem natural. A

Tabela 2 apresenta alguns dados dos textos selecionados, tais como, identificador do texto utilizado no instrumento, o domínio de aplicação do texto, o tamanho do texto (quantidade de termos) e a fonte de onde o texto foi extraído.

Para o processamento dos textos foi utilizada a aplicação LEO (*Learning Ontologies System*) [27] desenvolvida no escopo da pesquisa de Leão [26]. O LEO foi utilizado por apresentar uma abordagem mais eficiente para a identificação dos termos relevantes. O LEO integra ferramentas de processamento de linguagem natural, tais como: *StanfordNLP*, *OpenNLP*, *LTH Semantic Mapper* e *WordNet* (base semântica). 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.

**Tabela 2.** Textos selecionados para processamento

Id. Texto Instrumento/Texto	Domínio	Qtd. Termos	Referência
1.1	Comissão de Vendas	107	[22]
1.2	Biblioteca	213	[23]
1.3	Validação Serviços	243	[24]
2.1	Editores de Livros	76	[25]
2.2	Conferência Científica	103	[26]
2.3	Roteiro de Ônibus	329	[19]

Ao final do processamento automático, para cada descrição de domínio, uma lista dos termos identificados e seus respectivos construtos foram gerados. Foram elaborados dois instrumentos, cada um deles com o resultado obtido a partir de três descrições de domínio. Cada instrumento foi enviado para dezesseis especialistas em OntoUML. A lista de especialistas foi obtida por meio da seleção de publicações científicas relacionadas a OntoUML. Para apoiar a avaliação, quatro métricas foram estabelecidas: 1) percentual de termos relevantes corretos; 2) percentual de termos desambiguados corretos; 3) percentual de termos com consenso; e 4) percentual de acerto do método.

### 3.2 Avaliação da Derivação Automática dos Requisitos Funcionais

A partir dos resultados obtidos na etapa anterior, os modelos conceituais em OntoUML foram construídos semiautomaticamente. Nesta etapa, cinco modelos foram utilizados, todos da Tabela 1, exceto o Id 1.3 (devido a problema de qualidade do texto). Com os modelos conceituais construídos, foi executado o método de derivação apresentado na Seção 2.3. 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.

Para esta avaliação foram elaborados três instrumentos, os instrumentos 1 e 2 com duas descrições de domínio distintos e o instrumento 3 com uma descrição de domínio. O instrumento 3 foi elaborado com apenas uma 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. Por meio dos instrumentos, foi possível coletar as seguintes informações: 1) perfil dos participantes; 2) qualidade dos textos; 3) percentual de requisitos funcionais não corretos gerados pelo método proposto; 4) percentual de cobertura do método; e 5) percepções gerais da avaliação. Os instrumentos foram aplicados em dois grupos distintos: o primeiro, denominado profissionais, cujos participantes eram graduados e com experiência na atividade de levantamento de requisitos, e o segundo, denominado estudantes, cujos participantes estavam cursando graduação em computação que já haviam cursado disciplinas relacionadas a atividade de levantamento de requisitos.

## 4 Resultado da Avaliação

Esta seção apresenta os resultados referentes à avaliação da abordagem proposta. A avaliação foi conduzida em duas etapas: a primeira para a identificação automática dos elementos em um modelo conceitual OntoUML e a segunda para a derivação automática de requisitos funcionais a partir do modelo conceitual OntoUML.

### 4.1 Identificação Automática dos Elementos

Entre os especialistas (trinta e dois), dez retornaram com o instrumento preenchido. Cada instrumento (total de dois) foi respondido por cinco especialistas distintos. A Tabela 3 sumariza os resultados correspondentes às métricas estabelecidas para cada um dos seis textos avaliados.

Coluna (A) “termos relevantes corretos”: a média geral de acerto foi de 88% (variação de 80% a 100%). Não possível observar se o tamanho do texto interferiu no resultado, pois percentuais maiores foram obtidos tanto nos textos maiores quanto nos menores. Este resultado indicou um percentual de cobertura considerável. No instrumento, também foi solicitado que os especialistas indicassem termos relevantes não identificados pelo método. No total geral, apenas três termos foram indicados pelos especialistas como não identificados pelo método.

Coluna (B) “termos relevantes desambiguados corretos”: a média geral de acerto foi de 85% (variação de 75% a 94%). Esta medida não foi obtida por meio de especialistas porque não era necessário. A avaliação ocorreu manualmente pelo pesquisador comparando o significado retornado pelo algoritmo *TargetWord* [28] e o significado 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 não é feita de maneira correta, o método proposto pode fazer a indicação errada do construto. Também não foi possível encontrar relação entre o tamanho do texto e o percentual de acerto, pois percentuais maiores foram obtidos



em textos maiores e menores. 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 aconteceram antes do envio do instrumento para os especialistas. Este resultado reforça a necessidade de que ocorra uma análise manual do usuário na etapa de desambiguação para posteriormente seguir com as próximas etapas do método.

**Tabela 3.** Resultado das métricas para cada texto

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

Coluna (C) “consenso construto OntoUML”: a média geral de 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 pesquisa de que métodos devem ser desenvolvidos para apoiar o modelador na identificação dos elementos em um modelo conceitual em OntoUML.

Coluna (D) “acerto do método”: a média geral de 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 identificar as causas da falha do método. Entre os noventa e dois termos que obtiveram consenso entre os especialistas referente ao construto OntoUML correto, apenas para nove termos o método falhou. Entre esses termos, foi possível identificar as seguintes características: sete termos compostos não foram encontrados na base WordNet, um termo com o *supersense cognition* e um 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.

## 4.2 Derivação Automática dos Requisitos Funcionais

Nesta etapa da avaliação, participaram 83 pessoas, sendo 42 pessoas, categorizadas como profissionais, e 41 pessoas categorizadas como estudantes. O perfil de formação dos profissionais foi: 33% com especialização, 29% apenas graduação e 26% com mestrado. Entre os profissionais 85,7% afirmaram já terem atuado em atividades de levantamento de requisitos, com uma média de 8,5 anos de experiência nesta atividade. Entre os estudantes, 36,5% afirmaram já terem atuado em atividades de levantamento de requisitos, com uma média 1,3 anos de experiência nesta atividade. A avaliação foi realizada com dois públicos diferentes (estudantes e profissionais) para avaliar se a experiência ou não poderia interferir nos resultados.

A qualidade dos textos foi analisada segundo três critérios: 1) a facilidade de entendimento do domínio, 2) a clareza do texto e a 3) utilidade do texto com fins de levantamento de requisitos. De um modo geral, tanto para os profissionais, quanto para os estudantes, foram obtidos percentuais elevados de concordância. Nos três critérios, o percentual atingido, considerando as opções concordo e concordo totalmente, foram de no mínimo 68%. O critério que alcançou maior percentual foi em 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. Nenhum dos três critérios de qualidade apresentou resultado significativo que pudesse comprometer o resultado do método avaliado.

Em relação ao percentual de requisitos funcionais não corretos gerados pelo método proposto, em nenhum dos textos analisados, tanto por profissionais quanto por estudantes, se obteve percentual superior a 5%. Ou seja, os requisitos gerados pelo método, em sua maioria são corretos. A Figura 2 apresenta um recorte do Instrumento referente a lista de requisitos gerada pelo método para o texto “Comissão e Vendas”. Os itens gerados a partir do constructo “Role” (ex: RF 1.1.1; RF 1.1.1.1; RF 1.1.2), após a avaliação, foram suprimidos do método proposto pois não agregam com a lista de requisitos funcionais necessárias.

ID	Descrição do RFD	NÃO é um Requisito Funcional em um SI
<b>RF1</b>	<b>O sistema deve gerenciar Pagamento</b>	
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	
<b>RF2</b>	<b>O sistema deve gerenciar Venda</b>	
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 2. Recorte instrumento, lista de requisitos gerada (Texto Id. 1.1)

Os resultados referentes ao percentual de cobertura dos requisitos funcionais gerados pelo método proposto são apresentados na Tabela 4. Na tabela são apresentados os resultados por categoria de participantes (profissionais ou estudantes), a escala de cobertura que poderia ser selecionada pelo participante, e os percentuais obtidos para cada texto avaliado. Analisando os resultados individualmente (Tabela 4), é possível observar que foram obtidos percentuais diferentes para cada texto. Tanto na visão dos profissionais, quanto dos estudantes, os textos 2.2 (Conferência) e 2.3 (Roteiro de ônibus) foram os textos em que o método conseguiu maior percentual de cobertura. No texto 2.2, 92% dos profissionais e 77% dos estudantes concordaram que o método lista pelo menos 75% dos requisitos, enquanto no texto 2.3, 77% dos profissionais e 87% dos estudantes concordam que o método lista pelo menos 75% dos requisitos.

**Tabela 4.** Cobertura do método por texto

Categoria	Escala de cobertura	Id. Texto				
		1.1	1.2	2.1	2.2	2.3
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%

Na seção final do instrumento, foram apresentadas algumas afirmações para os participantes indicarem o percentual de concordância. Os resultados obtidos para estas afirmações foram os seguintes:

**Afirmação 1: 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 coerente. Acumulando o percentual entre “Concordo” e “Concordo totalmente”, obteve-se um percentual de 88% entre os profissionais e 76% entre os estudantes. Ou seja, a percepção geral dos participantes foi, em sua maioria, de que os resultados do método foram coerentes.

**Afirmação 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.** 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. Ou seja, a percepção geral dos participantes foi, em maioria absoluta, de que o método foi útil.

**Afirmção 3: 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. 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 demonstram que os participantes tiveram uma percepção bastante positiva com relação a coerência, a utilidade e mostram predisposição para usar a abordagem proposta. É importante ressaltar que esta etapa da avaliação foi conduzida com participantes mais experientes e menos experientes na atividade de levantamento de requisitos. No entanto, os resultados de ambas as categorias 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.

Como limitações desta pesquisa ressaltamos os seguintes aspectos: 1) as ferramentas de Processamento de Linguagem Natural, que produzem melhores resultados, processam textos em inglês, restringindo cenários em português; 2) os textos de domínio públicos encontrados são curtos e simples, haveria a necessidade de avaliação da proposta em textos mais longos e complexos; 3) a lista de requisitos do domínio gerada pelo método proposto é apenas uma relação em alto nível dos requisitos. Para um detalhamento destes requisitos, é necessário aprofundamento com o usuário.

## 5 Conclusão

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. Embora existam diversas técnicas para a elicitação de requisitos, algumas delas ainda são 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.

Esta pesquisa apresentou a avaliação de uma abordagem baseada em modelos conceituais, representada em OntoUML, para apoiar a derivação de requisitos funcionais de domínio. Os resultados obtidos apresentam evidências mais detalhadas sobre as possibilidades e os desafios da identificação semiautomática para a construção de um modelo conceitual. A avaliação envolveu seis textos de domínios diferentes com a participação de dez especialistas em OntoUML, contexto ainda não encontrado na

literatura. Os resultados também evidenciam como um modelo conceitual construído em linguagem mais expressiva pode apoiar a geração de requisitos funcionais de um software. O modelo conceitual pode contribuir para o entendimento mais próximo da realidade do domínio, antes de que artefatos tecnológicos pertinentes sejam desenvolvidos. Quanto maior 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.

Por fim, é importante ressaltar que estas conclusões foram obtidas considerando os contextos avaliados, e que para efeitos de generalização, novas avaliações em contextos distintos são necessárias.

## Referências

1. Gao, US General Accounting Office. Mission Critical Systems: Defense Attempting to Address Major Software Challenges, GAO/IMTEC-93-13, December (1992).
2. Zanolenc, 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).
3. Guizzardi, G. Ontological Foundations for Structural Conceptual Models, Telematica Instituut Fundamental Research Series 15, Universal Press (2005).
4. 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).
5. 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).
6. de la Vara, J.L., Sánchez J. Improving Requirements Analysis through Business Process Modelling: a Participative Approach, v. 1, p. 165–176 (2008).
7. 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).
8. Mylopoulos, J. Conceptual modeling and Telos, In P. Loucopoulos and R. Zicari, editors, Conceptual modeling, databases, and CASE. Wiley (1992).
9. Guizzardi, R., Franch, X., Guizzardi, G. 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).
10. 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, p. 72-77 (2015).
11. Valaski, J., Reinehr S., Malucelli, A. Retrospective and Trends in Requirements Engineering through the WER. In: Workshop em Engenharia de Requisitos, Montevideo (2013).

12. Valaski, J., Reinehr S., Malucelli, A. WER Overview: Retrospective, Trends and Relevance. *Clei Electronic Journal*, v.10, n. 3, (2013).
13. Kitchenham, B.A. Procedures for Undertaking Systematic Reviews, Joint Technical Report, Computer Science Department, Keele University (TR/SE0401) and National ICT Australia Ltda (2004).
14. 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, Las Vegas. p. 24-30 (2016).
15. Valaski, J., Reinher, S. and Malucelli, A. Evaluating the Expressiveness of a Conceptual Model Represented in OntoUML and UML. In: ONTOBRAS - Brazilian Ontology Research Seminar, Curitiba. ONTOBRAS (2016).
16. Valaski, J., Reinher, S. and Malucelli, A. Deriving domain functional Requirements from Conceptual Model Represented in OntoUML. In Proceedings of the 19th International Conference on Enterprise Information Systems. v. 2, p. 263-270 (2017).
17. Leão, F., Revoredo, K., Baião, F. Learning Well-Founded Ontologies Through Word Sense Disambiguation". In Proceeding Of: 2nd Brazilian Conference On Intelligent Systems (2013).
18. OLED. <https://code.google.com/p/OntoUML-lightweight-editor/>, acessado em: 2014 fev 11. (2014).
19. 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).
20. Valaski, J., Reinehr, S. and 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).
21. Valaski, J., Reinher, S. and Malucelli, A. Computational environment to Semi-Automatically Build a Conceptual Model Represented in OntoUML. In: ONTOBRAS - Brazilian Ontology Research Seminar, 2016, Curitiba. ONTOBRAS, (2016).
22. 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).
23. 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).
24. Deepthimahanti D. K., Sanyal, R. Semi-automatic generation of UML models from natural language requirements. In Proceedings of the 4th India Software Engineering Conference. ACM, p.165–174 (2011).
25. 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).
26. Leão, F. Expanding the Semantic Knowledge of Wordnet Through Semantic Types and UFO. Dissertação de mestrado. Universidade Federal do Estado do Rio de Janeiro (2014).
27. Leão, F. LEO - Learning Ontologies System. Acesso: 01/11/2016, disponível em: <https://github.com/felipeleao/leo>. (2016).
28. Pedersen, T., Kolhatkar, V. WordNet::SenseRelate::AllWords: a broad coverage word sense tagger that maximizes semantic relatedness. In Proceedings of Human Language Technologies. (2009).