

Uma abordagem para auxiliar estudantes com deficiência visual na modelagem de sistemas: um estudo piloto

Pedro O. de Azevedo¹, Vinícius S. L. Vieira¹, Alinne C. Correa Souza¹,
Francisco Carlos M. Souza¹

¹Coordenação de Engenharia de Software
Universidade Tecnológica Federal do Paraná – Dois Vizinhos – PR – Brasil

azevedo.2015@alunos.utfpr.edu.br, vsantos.lvieira@gmail.com

{franciscosouza, alinnesouza}@utfpr.edu.br

Abstract. *In recent years there has been a growing interest in attempts to include people, with special needs, performing Software Engineering activities such as programming and systems modeling. In this context, we present a tool called Blind Modeling system (B-Model) which aims to assist visually impaired people on system modeling activity supported by Unified Modeling Language (UML). For making the use of the B-Model approach feasible, we implemented a language called Blind Modeling Language (BML) and a prototype. The BML was evaluated with a blind student and the prototype was assessed through a pilot study using the requirements specification of two different scenarios performed by blind student for use case diagrams generation. The result shows the effectiveness of the B-Model approach, since the generated diagrams correspond to the specified functional requirements using the BML language.*

Resumo. *Nos últimos anos houve um interesse crescente em tentar incluir pessoas com necessidades especiais nas atividades de Engenharia de Software, como programação e modelagem de sistemas. Nesse contexto, neste artigo é apresentada uma ferramenta chamada Blind Modeling system (B-Model), que visa auxiliar pessoas com deficiência visual na atividade de modelagem de sistema utilizando a Linguagem de Modelagem Unificada. Para viabilizar o uso da abordagem B-Model foi desenvolvida uma linguagem chamada Blind Modeling Language (BML) e implementado um protótipo. A BML foi avaliada com uma estudante deficiente visual e o protótipo foi avaliado por meio de um estudo piloto usando a especificação de requisitos de dois cenários diferentes realizada pela estudante deficiente visual para geração de diagramas de casos de uso. O resultado mostra a eficácia da B-Model, pois os diagramas gerados correspondem aos requisitos funcionais especificados utilizando a linguagem BML.*

1. Introdução

Segundo o Instituto Brasileiro de Geografia e Estatística [IBGE 2010], 45 milhões de pessoas, ou seja, 23,9% da população total, apresentaram algum tipo de incapacidade ou deficiência. No que diz respeito à deficiência visual, atualmente, existem 6,5 milhões de brasileiros, sendo 582 mil cegos e 6 milhões com baixa visão. Neste contexto, a inclusão de estudantes com baixa visão ao ensino superior tem sido considerada desafiante.

Esse desafio é decorrente da baixa disponibilidade de livros e/ou materiais em formato acessível, leitores textuais, além das dificuldades no aprendizado de notações gráficas e a falta de capacitação de profissionais [Konecki et al. 2016].

Um obstáculo que estudantes com deficiência visual enfrentam no ensino superior é o aprendizado de disciplinas que exijam interpretações visuais [Giroto et al. 2012]. Dentre tais disciplinas é importante destacar Interação Humano Computador, programação para web, programação para dispositivos móveis e Análise Orientada a Objetos (AOO).

A disciplina de AOO visa demonstrar como é realizada a modelagem de sistemas utilizando a Linguagem de Modelagem Unificada (do inglês, *Unified Modeling Language - UML*). Essa modelagem apoia as especificações de um sistema por meio de diagramas, que permitem a compreensão das funcionalidades e comportamentos do software antes de desenvolvê-lo. Neste contexto, é importante destacar que os diagramas são utilizados com frequência para explicar conceitos, projetar sistemas ou mesmo documentar sistemas existentes. Diante deste cenário, os diagramas se tornam inacessíveis para os estudantes com deficiência visual devido à sua natureza gráfica.

Este artigo visa apresentar uma ferramenta chamada *Blind Modeling system (B-Model)*, a fim de auxiliar o aprendizado de estudantes com deficiência visual na modelagem de sistemas utilizando UML. A ferramenta é viabilizada por meio de uma linguagem chamada *Blind Modeling Language (BML)* para a especificação de requisitos, que a partir da qual possibilita gerar automaticamente o diagrama de caso de uso correspondente.

Este artigo está organizado da seguinte forma: na Seção 2 são apresentados os trabalhos relacionados. Na Seção 3 é detalhado a ferramenta B-Model. Na Seção 4 são apresentados a forma como foi conduzido o estudo de caso e experimento piloto. Na Seção 5 são apresentados e discutidos os resultados alcançados. Por fim, na Seção 6 as considerações finais e trabalhos futuros são apresentados.

2. Trabalhos Relacionados

A modelagem de sistemas é realizada por meio de diagramas que encapsulam informações visuais, compostos por formas geométricas e conectores relacionando-as. Diversos estudos tem investigado diferentes formas de auxiliar o ensino e a aprendizagem de modelagem de sistemas para deficientes visuais, tais como: ferramentas [Doherty and Cheng 2015], [King et al. 2004], [Pansanato et al. 2012]; representações físicas [Owen et al. 2014]; e notações textuais [Vieritz et al. 2012], [Grillo and de M. Fortes 2014] e [Loitsch et al. 2018], sendo esta última forma o foco deste artigo.

Apesar dos três estudos abordarem notação textual, somente dois [Harmain and Gaizauskas 2000] e [Elallaoui et al. 2018] utilizaram PLN para a geração de diagramas. No estudo de Harmain e Gaizauskas [Harmain and Gaizauskas 2000] foi desenvolvido um protótipo para geração do diagrama de classes a partir de uma especificação de requisitos em linguagem natural. Por outro lado, o trabalho de Elallaoui, Nafil e Touahni [Elallaoui et al. 2018] apresentou um processo de transformação de histórias de usuários em casos de uso utilizando técnicas de PLN juntamente com o analisador *TreeTagger*.

As diferenças da ferramenta B-Model em relação aos trabalhos apresentados são: (i) o foco no diagrama de caso de uso, uma vez que a partir deste é possível gerar o diagrama de atividades e por estar diretamente relacionado aos requisitos funcionais; (ii) o diagrama gerado está seguindo as notações da UML; e (iii) o uso da teoria de Chomsky [Chomsky 1956].

3. Ferramenta B-Model

Nesta seção serão delineados os detalhes da ferramenta B-Model cujo propósito é auxiliar estudantes com deficiência visual no aprendizado da modelagem de sistemas utilizando UML. Conforme pode ser visto na Figura 1, a B-Model é composta por três fases: (i) especificação dos requisitos funcionais em linguagem natural; (ii) interpretação dos requisitos; e (iii) geração do diagrama de caso de uso. Cada fase será detalhada nas próximas seções.

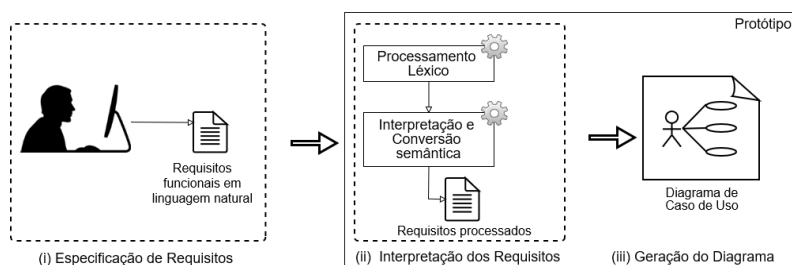


Figura 1. Visão geral da ferramenta B-Model

3.1. Fase 1: Especificação de Requisitos

Os requisitos de software representam as necessidades apresentadas pelos usuários ou clientes, que por sua vez devem ser contempladas pelo sistema a ser desenvolvido. Nesta fase o estudante deverá, em linguagem natural, especificar textualmente os requisitos de um cenário disponibilizado pelo professor, por exemplo.

Para o contexto deste artigo, somente serão considerados os requisitos funcionais, uma vez que estes são necessários para a geração do diagrama de casos de uso. As especificações dos requisitos serão escritas em português salvos em um arquivo de texto que será utilizado como entrada na próxima fase da B-Model.

3.2. Fase 2: Interpretação dos Requisitos

Para auxiliar a interpretação dos requisitos foi definida e desenvolvida uma linguagem chamada *Blind Modeling Language (BML)*. BML é uma linguagem que permite uma padronização na especificação dos requisitos para gerar o diagrama de caso de uso. Nesta fase é realizado o processamento e interpretação dos requisitos especificados utilizando a técnica PLN. A BML foi dividida em duas etapas: (i) processamento Léxico; e (ii) interpretação e conversão semântica.

No processamento léxico (etapa (i) da BML), os requisitos funcionais (entradas) são classificados em categorias conhecidos como *Parts Of Speech (POS)* [Allen 1995]. Nesta etapa foram consideradas as seguintes categorias:

- *determiners*: palavras que determinam o significado de um nome. Neste caso foram considerados os artigos [artigo] definidos no singular (O, o, a, A);

- *nouns*: representam os atores [ator], por exemplo usuário, os objetos [obj], por exemplo login, sistema. Ambos são substantivos e devem ser especificados sempre no singular.
- *verbs*: verbos no infinitivo, por exemplo realizar, que representam a ação que será executada no sistema.
- *prep*: representam as preposições para auxiliar a generalização entre casos de uso, por exemplo entre, em, com.

Além disso, algumas palavras-chaves foram definidas com base no domínio de caso de uso. Para isso, foram criadas três categorias especiais:

- *aux*: auxiliar na definição do nome do caso de uso por meio das palavras “deve” e “devem”;
- *conj*: conjunção para auxiliar na generalização de atores e de caso de uso, por meio das palavras “e”, “ou”, respectivamente;
- *dep*: representam os relacionamentos de *include* e *extend* por meio das palavras “incluindo” e “estendendo”, respectivamente;
- *símbolo*: representado por “;”;

Para a interpretação e conversão semântica dos requisitos funcionais foi utilizada a teoria de Chomsky [Chomsky 1956] que assume que as sentenças podem ser divididas em *Noun Phrase (NP)* e em *Verb Phrase (VP)*. Essa teoria foi aplicada devido utilizar a estrutura de sentenças como: sujeito, verbo e predicado; e facilitar a especificação de requisitos uma vez que é próxima da linguagem natural.

Além da teoria de Chomsky, regras foram definidas para auxiliar a interpretação e a conversão semântica. É importante ressaltar que a definição dessas regras é fundamental, pois permitem normalizar a escrita de uma sentença, ou seja, padronizar os requisitos especificados. A Tabela 1 apresenta as regras da BML utilizadas para geração do diagrama de caso de uso com base na especificação dos requisitos.

Tabela 1. Regras específicas da BML para a geração do diagrama de caso de uso com base na especificação dos requisitos

Elementos	Regras	Exemplo
Ator	[artigo] + [ator]	O usuário
Caso de uso	[verbo] + [obj]	realizar login
Requisito	[artigo] + [ator]+ [aux] + [verbo] + [obj]	O usuário deve realizar login.
Generalização de ator	[artigo] + [ator] + [conj] + [ator]+ [aux] + [verbo] + [obj]	O atendente e gerente devem realizar login.
Generalização de caso de uso	[artigo] + [ator] + [aux] + [verbo] + [obj] + [símbolo] + [verbo] + [prep] + [obj] + [conj] + [verbo] + [prep] + [obj]	O atendente deve realizar pagamento, pagar com dinheiro ou pagar com cartão.
<i>Include</i>	[artigo] + [ator]+ [aux] + [verbo] + [obj] + [dep] + [verbo] + [obj]	O atendente deve gerar relatório incluindo realizar login.
<i>Extend</i>	[artigo] + [ator]+ [aux] + [verbo] + [obj] + [dep] + [verbo] + [obj]	O atendente deve gerar relatório estendendo realizar impressão.

3.3. Fase 3: Geração do Diagrama

Para a geração do diagrama, o protótipo realizará o processamento léxico e a conversão semântica a partir dos requisitos funcionais especificados em língua portuguesa. É importante destacar que neste estudo os diagramas gerados contemplam atores, casos de uso, suas respectivas relações, e a generalização de atores.

O protótipo foi desenvolvido utilizando a linguagem python e duas bibliotecas dessa linguagem: (i) Spacy¹; e (ii) Tkinter². A biblioteca Spacy foi utilizada para realizar a etiquetagem das palavras em cada sentença buscando por partes do texto que representam os atores, os casos de uso e seus respectivos relacionamentos. Apesar da existência de outras bibliotecas similares, a seleção da *Spacy* foi devido permitir o processamento da língua portuguesa.

Após a etiquetagem das palavras, o processamento e a interpretação dos requisitos especificados são realizados utilizando PLN por meio da biblioteca *Spacy*. Cada parte de um requisito especificado é associado a um componente que irá compor o diagrama de caso de uso. Por fim, com a biblioteca *Tkinter* foi possível gerar o diagrama, uma vez que permite rotular os atores e casos de uso inserindo *labels* aos respectivos componentes.

4. Avaliação experimental

Dois estudos experimentais pilotos foram realizados: (i) estudo de caso para avaliar a utilidade da linguagem BML; (ii) experimento piloto para avaliar a eficácia da ferramenta B-Model. Em ambos os estudos foram utilizados as diretrizes propostas por Wholin et al. [Wholin et al. 2012].

4.1. Estudo de Caso

Este estudo foi realizado para avaliar a aceitação e uso da linguagem BML com uma estudante deficiente visual do curso de Bacharelado em Engenharia de Software durante a disciplina de AOO no período de agosto a dezembro de 2019. É importante destacar que algumas melhorias na BML foram realizadas durante o período da disciplina a partir dos *feedbacks* da estudante.

A avaliação da BML foi realizada com base no Modelo de Aceitação de Tecnologia (do inglês, *Technology Acceptance Model (TAM)* [Davis et al. 1989]. O modelo é baseado no pressuposto de que a intenção de utilizar um dispositivo tecnológico é determinada pela percepção do usuário sobre sua utilidade e sua facilidade de uso.

Após a definição e planejamento do estudo de caso, sua fase de condução foi realizada em duas etapas: (1) especificação de requisitos usando a BML e (2) análise da aceitação e uso da BML a partir dos requisitos especificados na primeira etapa. A primeira etapa visa especificar os requisitos de dois cenários diferentes ($C = c_1$ e c_2) utilizando as regras previamente estabelecidas na BML apresentada na Tabela 1. Esta etapa foi precedida por duas atividades. Na primeira atividade, o docente responsável pela disciplina explicou a linguagem para a estudante e criou um material com explicações e exemplos das regras da BML. Na segunda atividade, a descrição dos cenários foram disponibilizados à estudante, conforme apresentado na Tabela 2. Os requisitos de cada cenário foram especificados em um arquivo de texto.

Por fim, a etapa de análise da aceitação e uso está relacionada às respostas obtidas por meio de um questionário. O questionário foi elaborado com 11 questões em um arquivo de texto, as quais foram agrupadas em três categorias: (i) facilidade de uso da linguagem BML; (ii) utilidade da linguagem BML; (iii) intenção de usar a linguagem

¹Spacy documentation - <https://spacy.io/api/doc>

²Tkinter documentation - <https://docs.python.org/3/library/tk.html>

Tabela 2. Descrição dos dois cenários utilizados

c₁. Sistema de vendas de produtos eletrônicos: uma loja possui notebooks e celulares para venda. A loja possui um atendente cuja função é vender os produtos para os clientes e gerar relatórios semanais. A loja também possui um gerente cuja função é administrar o estoque e administrar fornecedores. Na ausência do atendente, o gerente também realiza as suas atividades. Além disso, após a compra o sistema gera a nota fiscal do produto.

c₂. Sistema de gerenciamento de festas: um sistema online deve gerenciar a compra e venda de ingressos de festa. Nesse sistema tanto o administrador como o gerente podem gerenciar as festas, gerar relatório e consultar festas. Além disso, o gerente pode adicionar patrocínios. O usuário do sistema realiza cadastro, compra ingresso e emite ingresso. Durante a compra do ingresso, a Administradora do cartão autentica a venda. Ao final da compra, o sistema envia um email com ticket do ingresso.

BML no futuro. A estudante realizou a avaliação da BML utilizando a escala *Likert* que consiste em cinco níveis, sendo o nível mais alto representado por “Extremamente” e o nível mais baixo “Nenhum pouco” [Likert 1932].

4.2. Experimento Piloto

Após a avaliação da BML, foi conduzido um experimento piloto para avaliar a eficácia da ferramenta proposta. Para a definição do experimento, o modelo *Goal-Question-Metric (GQM)* proposto por Basili e Weiss [Basili and Weiss 1986] foi utilizado para definir os objetivos deste experimento que consiste em: “*Analisar a ferramenta B-Model com o propósito de avaliar com respeito à eficácia do ponto de vista de pesquisadores no contexto de dois cenários diferentes.*”

Para alcançar o objetivo, a seguinte Questão de Pesquisa (QP) foi investigada: **O quão eficaz é a ferramenta B-Model para especificar requisitos e gerar o diagrama de caso de uso correspondente?** Para responder essa QP, a eficácia da abordagem foi mensurada por meio da relação do diagrama de caso de uso gerado com a especificação de requisitos utilizando a linguagem BML. Para o contexto deste experimento piloto, a primeira versão do protótipo foi avaliada somente com alguns os conceitos que contemplam o diagrama de caso de uso: (i) ator; (ii) caso de uso; (iii) associação; e (iv) generalização de atores. Portanto, os demais conceitos como *include*, *extend* e generalização de casos de uso serão incluídos na próxima versão do protótipo.

Para realizar a avaliação da ferramenta B-Model foram utilizados os dois cenários apresentados na Tabela 2. Apesar dos cenários serem fictícios, os mesmos compreendem os elementos necessários que um estudante deve abstrair para a modelagem de sistema. Por fim, para a condução do experimento piloto foram utilizadas como entrada as especificações dos requisitos dos cenários, realizadas pela estudante com deficiência visual utilizando a BML. Essas especificações foram inseridas no protótipo, as quais foram processadas e interpretadas para a geração do diagrama de caso de uso correspondente.

5. Análise e Discussão dos Resultados

Nesta seção são detalhados os resultados alcançados para a linguagem BML e para a ferramenta B-Model. Na Tabela 3 são apresentados os resultados das respostas da estudante em relação as categorias apresentadas na Seção 4.1. Os resultados sugerem que a BML é uma alternativa eficaz que pode ser utilizada para a especificação de requisitos e geração de diagrama de caso de usos. Um dos principais benefícios da BML é prover uma

especificação mais próxima da linguagem natural, garantindo assim, uma especificação mais consistente, inequívoca e completa para gerar os diagramas de caso de uso.

Tabela 3. Resultados referentes à facilidade de uso, utilidade e intenção de uso no futuro da BML

Categoria	Perguntas	Avaliação
Facilidade de uso	1. A sua interação com a linguagem BML foi clara e compreensível?	4
	2. Interagir com a linguagem BML exigiu muito esforço mental?	2
	3. Você considera a linguagem BML fácil de usar?	5
	4. Você considera a linguagem BML fácil de ser aprendida?	4
Utilidade	5. A linguagem BML melhorou seu desempenho?	5
	6. A linguagem BML permitiu aumentar sua produtividade?	5
	7. A linguagem BML foi útil para o aprendizado?	5
	8. A linguagem BML lhe permitiu realizar trabalhos em grupo?	5
	9. A linguagem BML pode lhe ajudar no mercado de trabalho?	5
Intenção de uso no futuro	10. Você pretende usar novamente a linguagem BML em outras disciplinas?	4
	11. Você recomendaria a linguagem BML para outras pessoas?	5

No que diz respeito ao experimento piloto, nas Figuras 2 e 3 são apresentadas as especificações de requisitos e os diagramas de caso de uso gerados para os cenários c_1 (sistema de vendas de produtos eletrônicos) e c_2 (sistema de gerenciamento de festa), respectivamente.

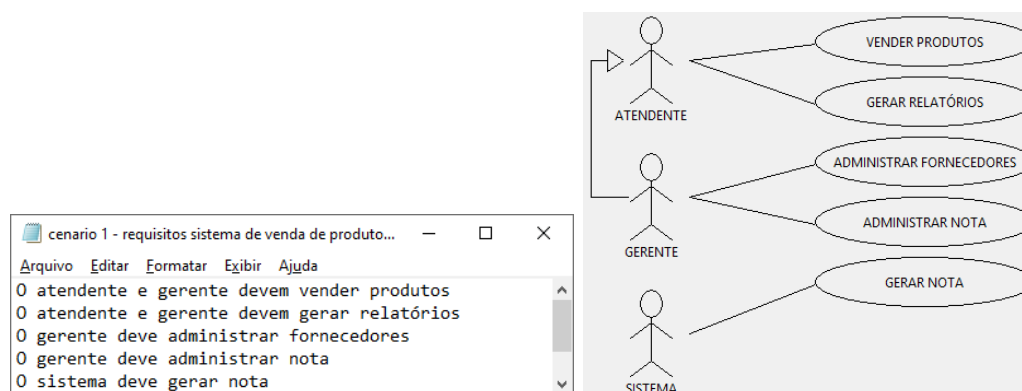


Figura 2. Especificação de requisitos do sistema de vendas de produtos eletrônicos e Diagrama de caso de uso correspondente gerado

É importante destacar que o enfoque deste experimento piloto foi corroborar se o protótipo gera o diagrama de caso de uso correspondente à especificação de requisitos utilizando a BML. Para isso, primeiramente, foi realizado um estudo de caso com uma estudante deficiente visual para a avaliação da BML. Nesse contexto, os resultados alcançados indicam que a ferramenta é eficaz, e consequentemente viável para auxiliar estudantes com deficiência visual na modelagem de sistemas. Conforme pode ser observado nas Figuras 3, por exemplo, a ferramenta B-Model foi capaz de reconhecer todos os atores e suas respectivas relações com seus casos de uso; diferenciou corretamente atores com identificadores similares, como é o caso de “Administrador” e “Administradora”; e atribuiu a generalização identificada aos atores que a possuem, considerando a sequência dos mesmos na especificação de requisitos.

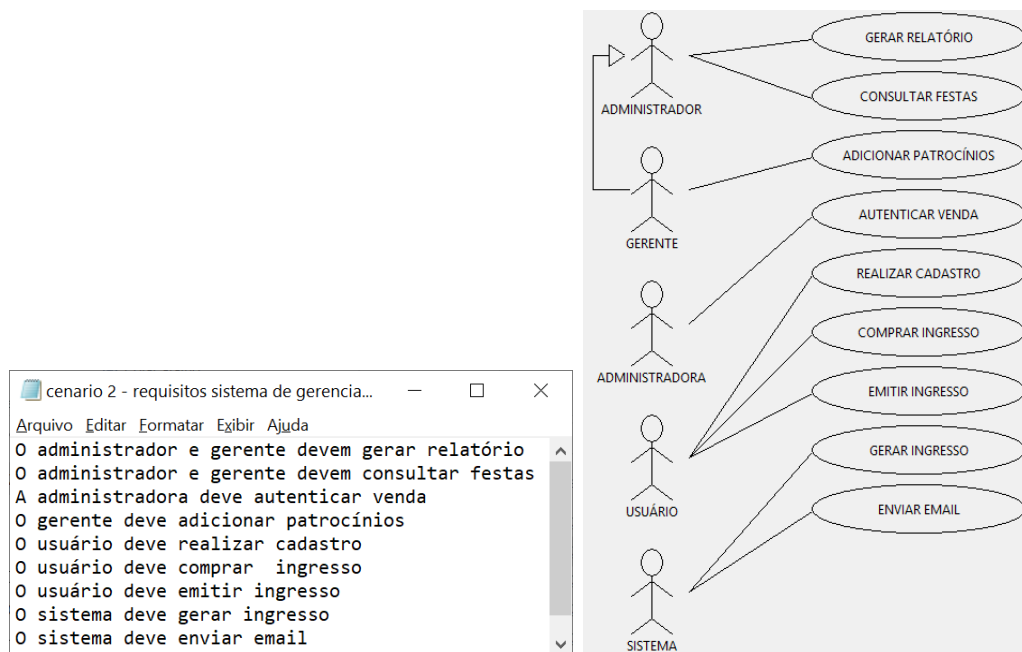


Figura 3. Especificação de requisitos do sistema de gerenciamento de festa e o seu respectivo diagrama de caso de uso gerado

No âmbito real de ensino e aprendizagem, por exemplo, estudantes com deficiência visual utilizam ferramentas de edição de texto para inserção do conteúdo e leitores de tela para verificação. Especificamente no contexto de modelagem de sistemas, o processo de ensino e construção de diagramas é realizado por meio de uma descrição textual, uma vez que estruturas visuais podem não fazer sentido ou nenhum sentido para estudantes que são deficientes visuais de nascença.

Portanto, não existe problema em realizar uma modelagem de sistemas por meio de uma descrição textual, pois é uma prática comum entre os estudantes com essa deficiência, seja no ensino, médio ou superior. No entanto, um artefato visual, como diagramas, é importante para inclusão destes estudantes em diferentes âmbitos, tais como:

- **ensino:** a modelagem de sistemas atualmente é realizada utilizando o padrão universal UML. Dessa forma, o artefato visual se torna imprescindível para a avaliação do estudante, bem como a sua interação com os demais colegas sem deficiência.
- **indústria:** assim como no ensino, os diagramas de UML são utilizados como documentos e artefatos para auxiliar no processo de desenvolvimento de software. Portanto, por meio de uma ferramenta de geração de diagramas via linguagem natural é possível que estudantes com deficiência visual tenham as mesmas oportunidades para ingressar em um cargo de engenheiro de software, por exemplo.

5.1. Ameaças a Validade

Nesta seção são detalhadas as possíveis ameaças à validade que podem afetar os valores e as conclusões do experimento piloto conduzido. As ameaças à validade externa estão relacionadas à generalização dos resultados. A representatividade dos cenários pode ser um problema, pois não existem teorias que garantam que um determinado conjunto

de cenários selecionados seja uma amostra representativa para a condução dos estudos empíricos. Com a finalidade de minimizar esse problema, foram utilizados dois cenários pertencentes a diferentes contextos. No entanto, não é possível afirmar que os resultados podem ser generalizados para qualquer cenário. Neste contexto, replicações futuras são necessárias para corroborar os resultados obtidos.

As ameaças à validade por construção estão preocupadas com a relação entre a teoria e o que é observado. No desenvolvimento da ferramenta proposta bem como na análise dos cenários, possíveis equívocos podem ter sido cometidos. Para mitigar esse risco foram realizados diversos testes das regras a fim de assegurar que durante a condução do experimento a linguagem e a ferramenta tivessem êxito na sua execução.

Por fim as ameaças à validade interna caracterizam o grau de confiabilidade entre os resultados esperados e os resultados obtidos. Todas as variáveis do experimento piloto foram controladas para mitigar possíveis ameaças. Além disso, para ampliar a confiabilidade dos resultados, os dados obtidos foram validados junto as regras definidas para assegurar que os resultados apresentados sejam realmente ortogonais, coerentes e interpretados de forma apropriada.

6. Considerações Finais

Neste artigo foram apresentados e discutidos os resultados da ferramenta B-Model que visa auxiliar estudantes com deficiência visual na modelagem de sistemas utilizando UML. Para a geração do diagrama de caso de uso, os requisitos funcionais especificados utilizando a linguagem BML foram processados por meio da técnica PLN juntamente com a teoria de Chomsky. Considerando a importância do diagrama de caso de uso, o foco que este recebeu ao longo do trabalho é justificado pelo fato de ser o principal diagrama utilizado no diálogo com o usuário ao identificar e validar os requisitos. Os casos de uso constituem elementos que estruturam todas as etapas do processo de software e, a partir deste, é possível gerar o diagrama de atividades, por exemplo.

Portanto, os resultados indicam que a ferramenta é promissora e eficaz pelas seguintes razões: (i) permite a descrição dos requisitos por meio de uma linguagem específica reduzindo alguns riscos presentes na especificação de requisitos como incompletude e ambiguidade; e (ii) gera automaticamente o diagrama de caso de uso baseado nos requisitos especificados utilizando a linguagem. Como trabalhos futuros, pretende-se: (i) avaliar a BML com outros estudantes deficientes visuais; (ii) incluir na ferramenta os relacionamentos de dependência do tipo *include* e *extend* e generalizações entre casos de uso; e (iii) validar a ferramenta com estudantes deficientes visuais.

Referências

- Allen, J. (1995). *Natural Language Understanding*. Addison-Wesley, 15th edition.
- Basili, V. R. and Weiss, D. M. (1986). A methodology for collecting valid software engineering data. *IEEE Transactions on Software Engineering*, 10(6):728–738.
- Chomsky, N. (1956). Three models for the description of language. *RE Transactions on Information Theory*, 2(3):113–124.
- Davis, F. D., Bagozzi, R. P., and Warshaw, R. P. (1989). User acceptance of computer technology: A comparison of two theoretical models. *Manage. Sci.*, 35(8):982–1003.

- Doherty, B. and Cheng, B. H. C. (2015). UML modeling for visually-impaired persons. In *Proceedings of the 1st International Workshop on Human Factors in Modeling co-located with 18th International Conference on Model Driven Engineering Languages and Systems*, pages 4–10.
- Elallaoui, M., Nafil, K., and Touahni, R. (2018). Automatic transformation of user stories into uml use case diagrams using nlp techniques. *Procedia Computer Science*, 130:42–49.
- Giroto, C. R. M., Poker, R. B., and Omote, S. (2012). *As tecnologias nas práticas pedagógicas inclusivas*. Editora Unesp, São Paulo.
- Grillo, F. D. N. and de M. Fortes, R. P. (2014). Accessible modeling on the web: A case study. *Procedia Computer Science*, 27:460–470.
- Harmain, H. M. and Gaizauskas, R. (2000). Cm-builder: an automated nl-based case tool. In *Proceedings in 15th IEEE International Conference on Automated Software Engineering*, pages 45–53.
- IBGE (2010). *Características Gerais da População, Religião e Pessoas com Deficiência*. IBGE, Rio de Janeiro, RJ.
- King, A., Blenkhorn, P., Crombie, D., Dijkstra, S., Evans, G., and Wood, J. (2004). Presenting uml software engineering diagrams to blind people. In *Proceedings of the International Conference on Computers for Handicapped Persons*, pages 522–529.
- Konecki, M., Ivković, N., and Kaniški, M. (2016). Making programming education more accessible for visually impaired. In *Proceedings of the 39th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO*, pages 887–890.
- Likert, R. (1932). A technique for the measurement of attitudes. *Journal Archives of Psychology*, 22(40):1–55.
- Loitsch, C., Müller, K., Seifermann, S., Henß, J., Krach, S., and Stiefelhagen, G. J. R. (2018). UML4ALL syntax – a textual notation for UML diagrams. In *Proceedings of the 16th International Conference on Computers Helping People with Special Needs, ICCHP'18*, pages 598–605. Springer.
- Owen, C. B., Coburn, S., and Castor, J. (2014). Teaching modern object-oriented programming to the blind: An instructor and student experience. In *Proceedings of the 121st American Society for Engineering Education, ASSE'14*, pages 1–13.
- Pansanato, L. T. E., Bandeira, A. L. M., dos Santos, L. G., and do Prado Pereira, D. (2012). Projeto D4ALL: acesso e manipulação de diagramas por pessoas com deficiência visual. In *Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems, IHC'12*, pages 33–36.
- Vieritz, H., Schilberg, D., and Jeschke, S. (2012). Access to uml diagrams with the HUTN. In *Proceedings of the 14th International ACM Sigaccess Conference on Computers and Accessibility, ASSETS'12*, pages 237–238.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in Software Engineering: An Introduction*. Springer-Verlag Berlin Heidelberg, 1st edition.