

UP₄EG: Um Perfil UML para Modelagem de Jogos Educacionais Digitais

Lucas R. Oliveira¹, Ana C. G. Inocêncio¹, Heitor Costa², Paulo A. Parreira Júnior²

¹ Instituto de Ciências Exatas - Universidade Federal de Goiás – Regional Jataí –
Caixa Postal 03 – 75801-615 – Jataí-GO – Brasil

² Departamento de Ciência da Computação - Universidade Federal de Lavras
Caixa Postal 3.037 – 37.200-000 – Lavras – MG – Brasil
{lukimrodrigues, anacarolina.inocencio}@gmail.com,
{heitor, pauloa.junior}@dcc.ufla.br

Abstract. *In this paper, we describe the UP₄EG, a UML profile that aims to facilitate the visualization, identification and understanding of the main elements of DEGs, from UML class diagrams. From an experimental study on the proposed profile, it was realized that when UP₄EG was used in the class diagrams of this type of software, the number of errors was significantly reduced.*

Resumo. *Este trabalho apresenta o UP₄EG, um perfil UML para facilitar a identificação, a visualização e o entendimento dos principais elementos de JEDs em diagramas de classes UML. A partir de um estudo experimental conduzido sobre o perfil proposto, percebeu-se que a quantidade de erros para identificação dos principais elementos de um JED pode ser significativamente reduzida, quando esse perfil é usado nos modelos de classes desse tipo de software.*

1. Introdução

Modelos de software são representações simplificadas de conceitos e objetos do mundo real, ressaltando suas principais características, de acordo com as necessidades do projeto (Sommerville, 2011). Modelos de softwares podem ser visualizados por meio de representações gráficas (Sommerville, 2011) e, de acordo com Fowler (2004), uma das linguagens mais conhecidas e utilizadas para representação gráfica de modelos de software é a UML (*Unified Modeling Language*).

Pelo fato de a UML ser uma linguagem genérica, isto é, que não foi criada para a representação de modelos de um tipo específico de software, os elementos de modelagem tradicionais dessa linguagem podem não ser suficientes para facilitar a identificação e o entendimento dos principais conceitos de domínios de software mais específicos (Gérard, 2011; Lisboa Filho *et al.*, 2010). Uma das alternativas propostas para contornar essa limitação é estender a UML por meio de um conjunto de mecanismos incluídos dentro de perfis UML (OMG, 2014). Segundo Lisboa Filho *et al.* (2010), um perfil UML é um conjunto de elementos que permitem alterar a sintaxe/semântica dos elementos da UML, sem violar sua semântica original.

Um exemplo de domínio de software bastante específico é o de Jogos Educacionais Digitais (JEDs). Uma modelagem tradicional, apenas com os elementos básicos da UML, pode não ser suficiente para representar tal domínio de forma clara. Isso faz com que possam surgir algumas dúvidas durante a modelagem, o entendimento e a manutenção de um JED, tais como: (i) quais elementos do jogo referem-se à

identidade do jogador? (ii) em quais componentes estão encapsuladas as fases do jogo? (iii) em quais classes se encontram os desafios? entre outros.

Neste trabalho, tem-se como hipótese que a identificação e entendimento dos principais elementos de um JED podem ser facilitados por meio da utilização de um perfil UML específico para esse domínio. Sendo assim, desenvolveu-se um perfil UML, denominado UP₄EG (*UML Profile for Educational Games*), composto de trinta e um elementos de modelagem, confeccionados com base na descrição dos principais elementos de um JED, proposta por Anetta (2010), bem como na modelagem de trinta e quatro jogos educacionais disponíveis na literatura e em repositórios de software livre. O perfil UP₄EG foi avaliado por meio de um estudo experimental, cujo intuito foi comparar a quantidade de erros ocorridos durante a identificação dos principais elementos de um JED utilizando o UP₄EG e outro perfil para modelagem de JEDs. Como resultados, percebeu-se que a quantidade de erros para identificação dos principais elementos de um JED pode ser significativamente reduzida, quando esse perfil é usado nos modelos de classes desse tipo de software.

O restante deste trabalho está organizado da seguinte forma: na Seção 2 são apresentados alguns conceitos básicos sobre JEDs e Perfis UML. Na Seção 3 estão os principais trabalhos relacionados ao tema desta pesquisa. Na Seção 4 encontra-se uma descrição detalhada do perfil UP₄EG, bem como da metodologia utilizada para sua confecção. Na Seção 5 estão o planejamento, a execução e a análise dos resultados de um estudo experimental conduzido sobre o perfil proposto. Por fim, na Seção 6 estão as considerações finais e as propostas para trabalhos futuros.

2. Background

2.1 Perfil UML

Um perfil UML é um conjunto de elementos que permitem a extensão da sintaxe e da semântica dos elementos pré-existentes na UML. Os principais elementos de um perfil UML são: *estereótipos*, *restrições* e *tagged values*. *Estereótipos* são como rótulos adicionados aos elementos de modelagem para destacar seu significado. Em geral, o nome de um estereótipo é representado entre os símbolos « e ». *Restrições* são responsáveis por aplicar regras aos estereótipos, definindo o que pode e o que não pode ser feito em relação à modelagem de um determinado domínio. *Tagged Values*, por sua vez, permitem definir informações adicionais a um determinado estereótipo.

Para exemplificar os conceitos apresentados anteriormente, suponha que em uma aplicação de *e-commerce*, os produtos do carrinho de compras do usuário são apresentados em uma tabela. Neste software, há uma classe denominada CarrinhoDeCompras, que corresponde a este carrinho de compras. Em um perfil UML, pode-se criar um estereótipo denominado «Tabela», que, quando colocado sobre o identificador da classe ListaDeCompras, ampliará o significado dessa classe, informando que, além de conter os produtos da compra de um usuário, ela será representada na interface do software por meio de uma tabela. Um possível exemplo de *tagged value* para o estereótipo «Tabela» seria *maxResultados*, cujo valor define a quantidade máxima de produtos a serem apresentados de cada vez ao usuário. Além do estereótipo e do *tagged value*, poderia ser criada uma restrição que impedisse o usuário de, durante a modelagem de um software com esse perfil, definir um valor para *maxResultados* menor do que 1 (um). Restrições, em geral, são representadas por meio da linguagem *Object Constraint Language* (OCL, 2016).

2.2 Jogos Educacionais Digitais

Jogos Educacionais Digitais (JEDs) são softwares aplicados ao ensino de determinados conteúdos a grupos específicos de aprendizagem. Anetta (2010) apresenta um conjunto de seis elementos considerados essenciais para um bom jogo: *identidade, imersão, interatividade, aumento de complexidade, ensino informado e didática*.

Identidade é o fator capaz de “fazer o jogador acreditar” que ele ou ela é um indivíduo dentro do ambiente do jogo. Em muitos jogos, a identidade dos jogadores é representada por meio de um único personagem, denominado *avatar*. **Imersão** significa que os jogadores têm um senso de presença por de sua identidade, estão envolvidos no conteúdo e, com isso, ganham motivação para alcançarem o objetivo do jogo. **Interatividade** significa que os jogos devem permitir que seus jogadores se comuniquem socialmente, tanto com outros jogadores (por exemplo, em um ambiente *multiplayer*), quanto com a máquina, comunicando-se com agentes computadorizados não controlados por humanos, também conhecidos como NPC (*Non-Player Character*). Segundo o conceito de **aumento de complexidade**, um bom jogo deve possuir diversos níveis. A conexão entre esses níveis em um jogo oferece uma plataforma para o aumento da complexidade dos conceitos em um jogo educacional. **Ensino Informado** refere-se à análise constante das informações dos estudantes no jogo (*feedback*). Isso oferece uma forma de se analisar o modo como os alunos estão lidando com os conceitos ensinados. O elemento **didática** consiste na forma como são dispostos os elementos educacionais necessários para o aprendizado do aluno. Eles devem ser organizados de uma maneira clara para que os jogadores tenham condições de identificar as atividades que devem ser realizadas em determinado momento do jogo.

3. Trabalhos Relacionados

Lisboa Filho *et al.* (2010) apresentam o “GeoProfile”, um perfil UML construído para a modelagem conceitual de bancos de dados geográficos. Segundo os autores, os modelos consolidados na literatura para a modelagem conceitual de banco de dados geográficos não abordavam todas as características consideradas relevantes para que se possa ter um entendimento adequado dos conceitos desse domínio. O trabalho de Torres *et al.* (2015), por sua vez, teve como objetivo a criação do perfil “MD-JPA”, que oferece uma extensão da UML para representação dos conceitos mais importantes sobre persistência de dados em Java, com base na especificação JPA (*Java Persistence API*).

Já no trabalho de Uetanabara Júnior *et al.* (2010), os autores propuseram o perfil “UML-AOF”, cujo objetivo era tornar as características dos *frameworks* Orientados a Aspectos (FOA) mais evidentes. De todos os trabalhos comentados, esse foi o único em que os autores realizaram uma avaliação do perfil proposto, comparando-o com outro perfil para modelagem de software orientado a aspectos, proposto por Evermann (2007). Os resultados dessa avaliação mostraram que, ao utilizar o perfil “UML-AOF”, menos erros foram cometidos durante a identificação dos principais elementos de um *framework* OO. No contexto da modelagem conceitual de jogos, Zagal *et al.* (2005) apresentam a ontologia GOP (*Game Ontology Project*), cujo objetivo é descrever e classificar os elementos essenciais de um jogo.

Dos trabalhos comentados anteriormente, nenhum deles tem como objetivo a confecção de um perfil UML para Jogos Educacionais Digitais. Quanto à ontologia GOP, apesar de seu intuito ser auxiliar no entendimento dos principais elementos de um jogo, seu uso não é totalmente adequado para a modelagem de JEDs, uma vez que

faltam elementos característicos do domínio educacional. Além disso, nenhum perfil UML foi proposto com base nessa ontologia, o que impede a utilização direta desses conceitos durante a modelagem de JEDs com a linguagem UML.

4. UP₄EG (UML Profile for Educational Games)

Para a criação do perfil UP₄EG, foi realizada a análise da documentação de trinta e quatro Jogos Educacionais Digitais (Tabela 1), com o intuito de se identificar como os elementos descritos na Seção 2 deste trabalho são modelados nesses jogos.

Tabela 1. JEDs utilizados como base para a confecção do perfil UP₄EG.

Categoria	Jogos Educacionais Digitais
Esporte	Billiard Game Physics (Curi, 2012), Ahunt (GitHub, 2015).
Estratégia	Abalone (GitHub, 2015), Dalek World (Duggan, 2015), Sokoban (Li, 2014).
Infantil	Eduka (GitHub, 2015), Ens. De Matemática (Mwangi, 2011), Ens. De Geografia (Lahoud, 2011), Contar é o Bicho (Mendes Junior, 2014), Heróis da Matemática (Mendes Junior, 2014), Qual letra é, Jacaré? (Mendes Junior, 2014), Maior ou Menor (Mendes Junior, 2014), Silabando (Mendes Junior, 2014), Tabuada Contra o Tempo (Mendes Junior, 2014), THOT (Teramoto, 2008).
Investigação	Jogo das Sete Falhas (Diniz, 2011), Mathemá (Aguar, 2012), Sim Investigador (Lessa Filho, 2014).
Quis	Fisicare (Jácome, 2003), Sócio-Histórico (Avelar, 2012), Multiagente (Vilela, 2011), SIEP (Silva, 2012), JOE (Paschoal, 2014).
Missão	ABC (Mendes, 2012), Macaco Acadêmico (Lobo, 2012), Treinamento para Gerentes de Projeto de Software (Dantas, 2013).
Simulação	CidTrans (Azzolini, 2014), Touck Brush Game (Rodrigues, 2011), Educação no Trânsito (Cuba, 2009), ITest Learning (Sousa, 2012), Percepção Musical (Matunobu, 2010), Tricky Trig (Tresohlavy, 2007), Monsterismus (GitHub, 2015), Reeducação Alimentar (Jardim, 2013), <i>Software Development Manager - SDM</i> (Kohwalter, 2011).

É importante salientar que perfis UML podem ser elaborados para qualquer tipo de diagrama disponibilizado na UML. Neste trabalho, escolheu-se confeccionar um perfil para o diagrama de classes da UML, pois, segundo Sommerville (2011), este é um dos diagramas mais utilizado para a modelagem de software em geral. Além disso, a documentação da maioria dos JEDs utilizados como base para a confecção do perfil proposto neste trabalho disponibilizava apenas seu diagrama de classes.

Para obtenção desses jogos, inicialmente, foi realizada uma busca em repositórios de código aberto, como *GitHub*, *Sourceforge*, entre outros. Foram encontrados quatro jogos nesses repositórios e, como nenhum deles disponibilizava seu diagrama de classes, foi realizada a confecção manual dos mesmos, a partir do código fonte disponível. Posteriormente, realizou-se uma busca por propostas de JEDs em repositórios de monografias, dissertações, teses e artigos científicos. A partir disso, trinta jogos educacionais foram encontrados, os quais já apresentavam um diagrama de classes elaborado pelos autores dos jogos.

A partir da análise dos trinta e quatro JEDs da Tabela 1, chegou-se a um conjunto de pacotes, estereótipos, *tagged values* e enumeradores (*enums*) que podem ser utilizados para a confecção de diagramas de classes UML de JEDs (Figura 1). Cada elemento de um bom JED, descrito na Seção 2, foi transformado em um pacote, que contém estereótipos relacionados a esse elemento. Os estereótipos do perfil proposto são representados como classes desse diagrama e seus *tagged values*, como atributos.

(i) **Identity** (Identidade) - nesse pacote está o conjunto de estereótipos relacionados à identidade de um jogador; o estereótipo «Identity» é responsável por destacar em qual classe está modelado o *avatar* que representa o jogador. Por meio do *tagged value* *protagonist*, o engenheiro de software pode modelar o tipo de *avatar* que representará o jogador. Esse tipo é definido pela enumeração *ProtagonistType*, que contém os seguintes valores: *Hero*, *Warrior* e *Animal*. Além disso, um *avatar*

poderá apresentar habilidades no jogo, representadas pelo estereótipo «Skill»; essas habilidades são divididas em dois tipos: (i) habilidades temporárias, representadas pelo estereótipo «TemporarySkill»: são consideradas habilidades necessárias para que o jogador conclua pequenos objetivos dentro do jogo. Por exemplo, um *avatar* do tipo guerreiro, que precisa derrotar um tipo diferente de vilão em cada fase, e, para isso, necessita de uma habilidade diferente em cada uma; e (ii) habilidades permanentes, representadas pelo estereótipo «PermanentSkill»: são consideradas habilidades em que uma vez que conquistadas o jogador não as perde.

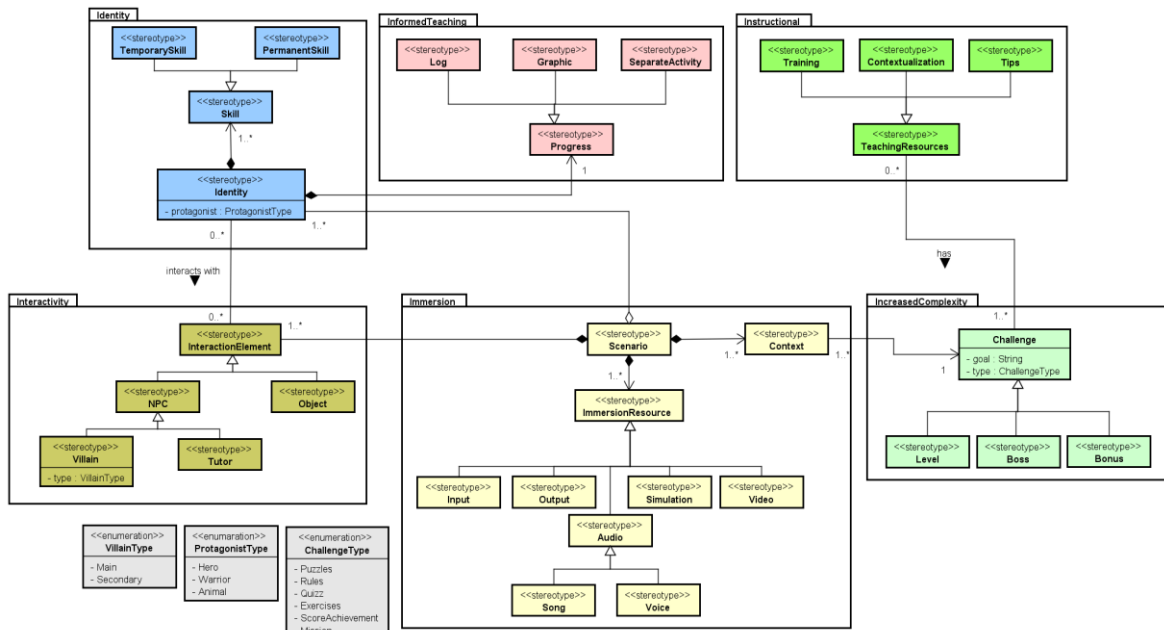


Figura 1. O perfil UP₄EG.

(ii) **Interactivity** (Interatividade) – esse pacote representa o conjunto de estereótipos responsáveis por dar significado aos elementos do jogo que irão interagir com o jogador. Sendo assim, há um relacionamento entre os estereótipos «Identity» e «InteractionElement». Os elementos de interação são representados por dois estereótipos: (i) «NPC», que possui como subclasses, os estereótipos «Villain» e «Tutor». «Villain» representa os vilões do jogo; como em um jogo podem existir diversos vilões, tal informação é representada no perfil por meio do *tagged value* type e da enumeração `VillainType`, cujos possíveis valores são `Main` e `Secondary`. Por exemplo, no jogo “Macaco Acadêmico” (Lobo, 2012), o jogador deverá coletar frutas para enfrentar diversos monstros existentes no jogo, que podem ser representados pelo estereótipo «Villain» com o valor `Secondary` para o atributo `type`. Contudo, há um vilão principal em cada fase do jogo; e (ii) «Object», que representa os objetos no mundo do jogo, os quais serão passíveis de interação com o jogador, como por exemplo, uma pedra, uma árvore, uma espada, entre outros.

(iii) **Immersion** (Imersão) – nesse pacote estão os estereótipos responsáveis por caracterizar os elementos relacionados à imersão. O estereótipo «Scenario» pode ser usado para modelar os elementos do jogo que caracterizam o cenário em que se desenrolará todos os acontecimentos do jogo. Um elemento modelado como «Scenario» pode conter diversos elementos do tipo «InteractionElement» e «Identity» (se for considerada uma plataforma *multiplayer*). Como pode-se perceber na Figura 1, o estereótipo «Scenario» possui relacionamentos de composição com

alguns dos estereótipos inerentes à representação dos elementos de JEDs. Uma composição representa um relacionamento do tipo todo-parte, em que o tempo de vida dos objetos-parte depende do tempo de vida do objeto-todo. Por exemplo, se um cenário for removido, seus elementos de interação também serão removidos. Além disso, um objeto-parte não pode estar relacionado a mais de um objeto-todo simultaneamente. Isso faz sentido, pois um elemento de interação pertence a um e apenas um cenário a cada vez. Além disso, cada cenário pode estar relacionado a um ou mais elementos do tipo «Context». Por exemplo, no jogo “CidTrans” (Azzolini, 2014), cujo objetivo é apoiar o ensino de leis de trânsito, o jogador deve dirigir um carro em um tráfego simulado. Cada fase do jogo representa um cenário diferente (uma cidade de pequeno, médio ou grande porte), mas em cada cenário, o jogador poderá concluir diferentes desafios (representados pelo estereótipo «Challenge»), de acordo com o contexto atual daquela fase, ou seja, em cada fase poderá haver contextos distintos, com também desafios distintos.

Além disso, o cenário deve ter alguns mecanismos que irão permitir saídas de áudio e vídeo ao usuário, além de suportar a entrada de dados e permitir um *feedback* de acordo com a entrada realizada. Todos esses elementos contribuem para a imersão do jogador e são representados por meio das subclasses do estereótipo «ImmersionResource»: (i) «Input» e «Output» são responsáveis pela entrada e saída de dados no jogo; (ii) «Simulation» tem como objetivo representar simulações existentes em um jogo. Por exemplo, o jogo “Fisicare” (Jácome, 2003), que possui como objetivo auxiliar no ensino de física, permite ao aluno visualizar simulações de experimentos físicos; (iii) «Animation» é utilizado para representar uma pequena história do jogo ao jogador, com o propósito de introduzi-lo no contexto do jogo. Isso é perceptível no jogo “Sócio-Histórico” (Avelar, 2012), em que são apresentadas telas com o intuito de contextualizar o jogador diante de uma história; e (iv) «Audio» é responsável pela representação dos sons do jogo, sendo eles músicas de fundo («Song») ou vozes («Voice»), tanto de personagens como de narradores.

(iv) **IncreaseComplexity** (Aumento de complexidade) – o estereótipo «Challenge», contido nesse pacote, é responsável por representar os tipos de desafios abordado no jogo, bem como o objetivo dos mesmos, respectivamente modelados pelos seus *tagged values type* e *goal*. Além disso, os estereótipos «Level», «Boss» e «Bonus» são tipos de desafios que podem ser melhor explicitados por meio do enumerador «ChallengeType». Por exemplo, em um jogo, poderia haver um desafio do tipo “avançar de fase”, que corresponderia ao estereótipo «Level». Mas esse desafio (passar de fase) poderia exigir a realização de um quebra-cabeça (*puzzles*) ou finalização de uma missão (*mission*) dada ao jogador. Assim, o *tagged value type* pode auxiliar na modelagem desse cenário. O *tagged value goal*, por sua vez, representa um texto que descreve o objetivo do desafio, como por exemplo, “finalizar a missão no menor tempo possível”, “finalizar o quebra-cabeça com o menor número de erros”, entre outros.

(v) **Instructional** (Didática) – o pacote Instructional é responsável por representar o conjunto de estereótipos capazes de explicitar quais são os elementos de aprendizagem a serem abordados no jogo. Tais objetos instrucionais são representados por meio do estereótipo «TeachingResources», que tem relacionamento com estereótipo «Challenge». Isso se justifica, pois como os desafios são uma das principais características para a imersão do jogador, quanto mais envolvido o aluno

estiver, mais ele estará aprendendo novos conteúdos e fixando os já obtidos. Contudo, nem todo desafio precisa estar diretamente relacionado a um elemento instrucional.

O estereótipo «TeachingResources» possui três subclasses: (i) «Training», que representa elementos instrucionais do tipo treinamento. Geralmente são usados quando o jogador não possui familiaridade com o conteúdo instrucional a ser abordado no jogo; (ii) «Contextualization», responsável por contextualizar (problematizar) o conteúdo a ser contemplado no jogo. Esta característica pode ser percebida, por exemplo, nos jogos “SDM” (Kohwalter, 2011), “Jogo das Sete Falhas” (Diniz, 2011) e “Treinamento para Gerentes de Projeto de Software” (Dantas, 2013); e (iii) «Hints», que representa as dicas que são oferecidas durante o jogo para auxiliar no cumprimento de algum desafio. No jogo “Jogo Ortográfico Educacional – JOE” (Paschoal, 2014), por exemplo, quando o aluno comete algum erro, são apresentadas dicas para possíveis soluções da atividade.

(vi) **InformedTeaching** (Ensino informado) – nesse pacote estão os estereótipos responsáveis por representar o desempenho do aluno no jogo. O principal estereótipo desse pacote é «Progress», que possui três subclasses: (i) «Log»: representa as ações do aluno no jogo, geralmente armazenada em arquivos de *log*; (ii) «Graphic»: representa os resultados obtidos pelos alunos no jogo por meio de gráficos, de forma a fornecer ao professor uma interpretação dos dados de forma mais clara, rápida e objetiva; e (iii) «SeparateActivity»: representa o desempenho dos alunos em pequenas atividades no jogo como por exemplo, número de respostas corretas em uma fase, número de moedas coletadas, quantidade de pontos obtidos, entre outras coisas.

Na Figura 2 é apresentado um trecho do diagrama de classes do jogo “SDM”, modelado com o auxílio do perfil UP₄EG. SDM é um acrônimo para *Software Development Manager* e representa um jogo educacional para a disciplina “Engenharia de Software”, cujo objetivo é auxiliar no ensino de conceitos e práticas relacionadas à gestão de pessoas, incluindo capacitação, carga de trabalho e papéis e responsabilidades de cada empregado em um ambiente de desenvolvimento de software (Kohwalter, 2011). Na Figura 2, a classe Employee é responsável por representar os empregados do jogo, bem como os atributos de cada empregado, incluindo o jogador. Por representar a identidade do jogador, ela encontra-se estereotipada com «Identity».

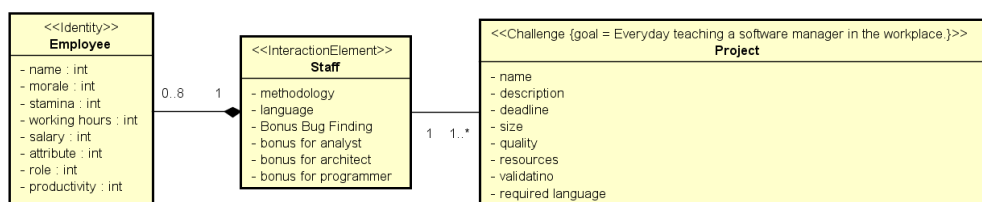


Figura 2. Parte do diagrama de classes UML do jogo “SDM”.

Além disso, a classe Employee compõe a classe Staff, responsável por representar os funcionários da equipe de desenvolvimento. Ao ser aplicado o estereótipo «InteractionElement», esta classe passa a oferecer novas informações sobre o jogo ao Engenheiro de Software, explicitando que esta classe também possui como característica a representação dos elementos de interação entre o jogador e os demais funcionários da equipe de desenvolvimento. O estereótipo «Challenge» foi aplicado à classe Project, que está associada à classe Staff, representando que o desafio do jogo se encontra modelado nesta classe. Essa classe representa o projeto de software que

deve ser executado pelo jogador, juntamente com a equipe de desenvolvimento, em um prazo (*deadline*) pré-estabelecido e com um nível de qualidade aceitável.

5. Estudo Experimental

O estudo experimental descrito nesta seção teve como objetivo verificar a eficácia da aplicação do perfil UP₄EG quanto à correta identificação dos elementos de um JED. No contexto desse trabalho, eficácia está relacionada à quantidade de respostas corretas dadas pelos participantes do experimento, a perguntas que analisam o seu entendimento com relação aos elementos de um JED modelados em um diagrama de classes UML.

O perfil proposto foi comparado com um perfil UML desenvolvido pelos autores desse trabalho, a partir da ontologia para jogos (GOP), proposta por Zagal *et al.* (2005). Isso foi necessário, pois não havia outros perfis UML similares ao proposto neste trabalho para comparação. Daqui adiante, esse perfil será referenciado como Perfil GOP. Devido à limitação de espaço, o perfil GOP não foi apresentado neste artigo, mas o mesmo pode ser encontrado em Oliveira (2016).

5.1. Planejamento do Estudo Experimental

A avaliação foi realizada com 10 (dez) graduandos do curso de Ciência da Computação. Diagramas de classes de dois jogos educacionais foram escolhidos, dentre os trinta e quatro diagramas de classes analisados neste trabalho. A escolha foi feita de forma que o diagrama de classes dos dois jogos possuísse uma quantidade aproximada de classes e elementos relacionados aos JEDs.

Com o intuito de homogeneizar o conhecimento dos avaliadores, foi realizado um **treinamento** sobre modelagem de software, diagramas de classe UML e sobre os perfis UP₄EG e GOP. Após isso, os participantes do experimento foram divididos em dois grupos de cinco, a saber, *Grupo A* e *Grupo B*. Durante a **primeira fase** do estudo experimental, o *Grupo A* recebeu o diagrama de classes do jogo “CidTrans” (Azzolini, 2014), modelado com base no perfil UP₄EG e o *Grupo B* recebeu o diagrama de classes do mesmo jogo, modelado de acordo com o Perfil GOP. Na **segunda fase**, o *Grupo B* recebeu o diagrama de classes do jogo “Sim Investigador” (Lessa Filho, 2014), modelado com auxílio do perfil UP₄EG, e o *Grupo A* recebeu o mesmo diagrama, porém modelado com o Perfil GOP.

Em cada fase do experimento, foi entregue um questionário aos participantes, que deveria ser respondido após a análise dos diagramas de classes apresentados a eles. Tal questionário continha seis questões relacionadas à identificação e à compreensão dos elementos de um JED modelados nesses diagramas, tais como identidade do jogador, agentes responsáveis pela interação com o jogador, entre outros. O questionário aplicado para a avaliação não pode ser totalmente apresentado neste artigo por falta de espaço, porém ele pode ser encontrado no trabalho de Oliveira (2016). As respostas dadas pelos participantes do experimento foram corrigidas e analisadas por especialistas nos assuntos de modelagem de software e jogos educacionais.

5.2. Resultados e Discussão

Os gráficos das Figuras 3 e 4 resumem os resultados obtidos na primeira e segunda fases do experimento, respectivamente. Esses gráficos apresentam a porcentagem de respostas incorretas, por questão, dadas pelos participantes que utilizaram diagramas modelados com os perfis analisados, a saber, UP₄EG e GOP.

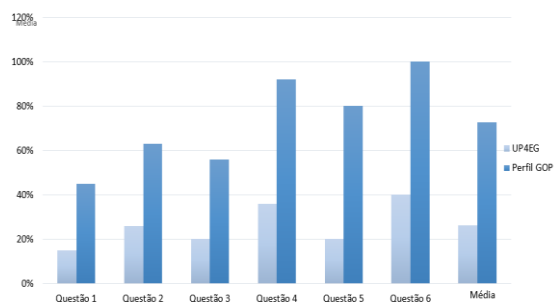


Figura 3. Resultados - 1ª fase

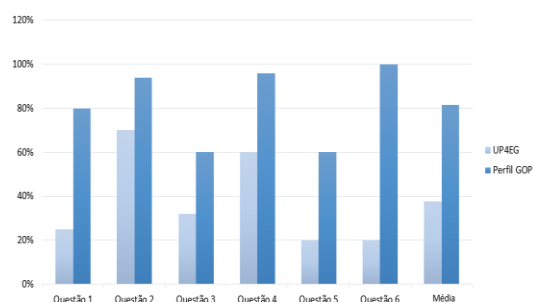


Figura 4. Resultados - 2ª fase

Um resultado que chamou a atenção, devido às discrepâncias existentes entre os valores para os dois perfis, é o da **Questão 5**, na primeira fase do experimento. Essa questão pede ao participante que informe se o jogo possui mecanismos para escolha de seu grau de complexidade e a pede que ele aponte quais elementos do diagrama são responsáveis por representar esses mecanismos. A resposta esperada para essa questão seria que o jogo provê tal escolha e que a mesma é representada pela classe *Cenário*. Apesar de a ontologia GOP tratar da modelagem conceitual de jogos, o elemento “aumento de complexidade” é um conceito importante, principalmente, em jogos educacionais. Sendo assim, tal perfil não possibilitou a correta identificação desse conceito para a maioria dos participantes. Quanto à segunda fase do estudo experimental, a **Questão 6**, relacionada à identificação dos elementos responsáveis pelo *feedback* do progresso do aluno no jogo, foi a questão em que houve maior discrepância entre os dois grupos. De modo análogo ao que ocorreu no exemplo anterior, o perfil GOP não ofereceu recursos adequados para a representação desses tipos de elemento, o que dificultou a resolução dessa questão por parte dos participantes.

Em resumo, a quantidade média de erros cometidos na primeira fase do experimento pelos participantes que utilizaram o perfil UP₄EG foi consideravelmente menor do que a quantidade de erros dos participantes que utilizaram o Perfil GOP. Isso ocorreu, pois o perfil UP₄EG pôde ser aplicado em todas as classes do jogo “CidTrans”, permitindo assim, a ampliação das informações presentes nessas classes. A quantidade média de erros na segunda fase do estudo experimental, cometida pelo grupo que utilizou o perfil UP₄EG, também foi menor. Porém, mesmo utilizando esse perfil, ainda houve erros. Uma explicação para isso pode ser a falta de conhecimento dos participantes quanto aos jogos apresentados a eles, bem como a ambiguidade gerada pela nomenclatura de alguns estereótipos desse perfil. Sendo assim, pretende-se replicar o experimento descrito nesta seção com outros participantes e com outros jogos, com o intuito de se obter resultados mais robustos.

6. Considerações Finais

Este trabalho apresentou detalhes da confecção de um perfil UML para a modelagem de classes de jogos educacionais digitais. Esse perfil foi elaborado com base em trinta e quatro jogos previamente selecionados na literatura e em repositórios de software livre. Os resultados do estudo experimental conduzido sobre o perfil UP₄EG, e detalhado neste artigo, mostraram que há indícios de que esse perfil pode facilitar a identificação e entendimento dos principais elementos de um JED, por parte de seus desenvolvedores, a partir de um diagrama de classes UML. Como trabalhos futuros, pretende-se: (i) implementar o perfil UP₄EG em alguma ferramenta de apoio à modelagem de software, como o Astah*; (ii) incrementar o perfil proposto com novos elementos, que podem ser

obtidos a partir da análise de outros tipos de jogos; e (iii) propor a utilização do UP₄EG para a detecção de problemas de projeto de JEDs logo no início do desenvolvimento dos mesmos.

Referências

- Aguiar, T. e ALEXANDRE, E. O Uso de Jogos Educativos no Ensino da Matemática Elementar. 34 f. Monografia (Bacharelado em Sistemas para Internet) - Centro Universitário Adventista de São Paulo, Campus Engenheiro Coelho - São Paulo. 2012.
- Anneta, L. A. The "I's" Have It: A Framework for Serious Educational Game Design. American Psychological Association. 2010.
- Avelar, F. T. Desenvolvimento de um jogo educacional de cunho sócio-histórico. 58 f. Monografia (Bacharelado em Ciência da Computação) - Universidade Estadual Paulista - Faculdade de Ciências, Bauru - São Paulo. 2012.
- Azzolini, V. L. CIDTRANS - Jogo 3D para educação no trânsito. 2014. 99 f. Monografia (Engenharia da Computação) - Universidade Tecnológica Federal do Paraná, Pato Branco. 2014.
- Booch, G. *et al.* The Unified Modeling Language User Guide, 2ª edição. Rio de Janeiro, Elsevier, 2005. 463 p.
- Cuba, G. P. Jogos sérios: tecnologia de jogos por computador aplicada ao ensino e aprendizagem. 91 f. Monografia (Bacharelado em Ciência da Computação) - Universidade Federal de Goiás - Campus Catalão. 2009.
- Curi, C. F. *et al.* Proposal and Development of a Computer Game for Teaching Physics. SBC - Proceedings of SBGames. 2012.
- Dantas, A. R. Jogos de Simulação no Treinamento de Gerentes de Projetos de Software. Dissertação (Mestre em Ciências em Engenharia de Sistemas e Computação) - Universidade Federal do Rio de Janeiro, Rio de Janeiro. 2013.
- Diniz, L. L. Jogo das 7 Falhas: Um Jogo Educacional para o Apoio ao Ensino do Teste de Caixa-Preta. Dissertação (Mestre em Computação Aplicada) - Universidade do Vale do Itajaí, São José - Santa Catarina. 2011.
- Duggan, Brian *et al.* Learning games programming with Dalek World. 7th. International Conference on Computer Games: AI, Animation, Mobile, Educational and Serious Games, CNBDI, 28-30th. November, Magelis, Angouleme, France. 2015.
- Evermann, J. A Meta-Level Specification and Profile for AspectJ in UML. Journal of Object Technology. Vol 6, No 7, Aug 2007.
- Fowler, M. UML Essencial: Um breve guia para a linguagem-padrão de modelagem de objetos, São Paulo, Ed. S.A, 2004. 155 p.
- Gérard, S. Papyrus User Guide Series: About UML profiling. 2011. Disponível em: https://www.eclipse.org/papyrus/resources/PapyrusUserGuideSeries_AboutUMLProfile_v1.0.0_d20120606.pdf
- Jardim, R. S. Jogo Virtual de Reeducação Alimentar Infantil. 78 f. Monografia (Bacharelado em Sistemas de Informação) - Universidade Federal do Estado do Rio de Janeiro - Centro de Ciências Exatas e Tecnologia, Rio de Janeiro. 2013.
- Jácome, T. F. FISICARE - Um Software Educacional Para Auxílio à Aprendizagem em Física (Temperatura e Calor). 45 f. Relatório de Estágio (Sistemas de Informação) - Centro Universitário Luterano de Palmas, Palmas. 2003.
- Uetanabara Júnior, J. *et al.* An Overview and an Empirical Evaluation of UML-AOF. ACM SAC 2010.
- Lahoud, Y. M. S. *et al.* Jogo Educativo para a Disciplina de Geografia. Colloquium Exactarum, v. 3, n. 2, p. 133-145. 2011.
- Lessa Filho, C. A. *et al.* Um Jogo Educativo na Web no Contexto do Ensino Fundamental. CINTED - Novas Tecnologias na Educação, V. 12 Nº 2, dezembro, 2014.
- Lisboa Filho, J. *et al.* (2010). A UML profile for conceptual modeling in GIS domain. DE@CAISE'2010, p.18-31.
- Lobo, A. S. e Mello, L. S. V. Desenvolvimento de um jogo para apoio ao ensino de aritmética. 53 f. Monografia (Bacharel em Ciência e Tecnologia) - Instituto Federal de Educação, Campus Guarulhos. 2012.
- Matunobu, Y. Desenvolvimento de um Software Educativo para o Treinamento em Percepção Musical. 81 f. Monografia (Bacharelado em Ciência da Computação) - Centro Universitário Eurípides De Marília - UNIVEM, Marília. 2010.
- Mendes Júnior, A. Z. ABC - Sistema de Auxílio à Educação Infantil. 2012. Monografia - IFP - Campus Paranaguá. 2012.
- Mwangi, Rose W. *et al.* Integrating ICT With Education: Designing an Educational Computer Game for Teaching Functions in Undergraduate Mathematics. Journal of Theoretical and Applied Information Technology. 2011.
- Object Constraint Language (OCL). Disponível: <http://www.omg.org/spec/OCL/>. Acessado em: 22 mar. 2016.
- Object Management Group (OMG). Meta Object Facility (MOF) Core Specification. Disponível: <http://www.omg.org/spec/MOF/2.4.2/PDF>. Documento: formal/2014-04-03.
- Paschoal, Luan *et al.* JOE: Jogo Ortográfico Educacional. III Congresso Brasileiro de Informática na Educação (CBIE 2014). XXV Simpósio Brasileiro de Informática na Educação (SBIE 2014).
- Rodrigues, H. F. Aplicando Sistemas Hápticos em Serious Games: Um Jogo para a Educação em Higiene Bucal. 156 f. Dissertação (Mestrado em Informática) - Departamento de Informática - Universidade Federal da Paraíba, João Pessoa. 2011.
- Silva, D. O. SIEP: Sistema Interativo Educacional Pedagógico. Monografia. Faculdade Cenecista de Capivari. São Paulo. 2012.
- Sommerville, I. Engenharia de Software. 9ª Edição. São Paulo: Pearson Prentice Hall, 2011. 529 p.
- Sousa, V. F. ITest Learning: Um jogo para o apoio ao ensino de testes de software. Monografia (Bacharel em Sistemas de Informação) - Universidade Federal do Ceará, Campus Quixadá. 2012.
- Teramoto, E. H. I. *et al.* TOTH: Jogo Eletrônico para a Aprendizagem da Matemática. 80 f. Monografia (Engenharia da Computação) - Universidade de São Paulo - Escola Politécnica, São Paulo. 2008.
- Torres, A. *et al.* Towards a UML profile for model-driven object-relational mapping. SBES. 2009.
- Tresohlavy, K. Educational Game: Tricky Trig. Object-Oriented Theory and Programming I. 2007.
- Vilela, J. F. F. Projeto e implementação de um Software Educativo Multi-Agente com Tropos e JADE. 129 f. Monografia (Engenharia da Computação) - Universidade Federal do Vale do São Francisco - UNIVASF, Juazeiro - Bahia. 2011.
- Zagal, José P. *et al.* Towards an Ontological Language for Game Analysis. 2005. Disponível em: <http://www.gameontology.com/>.