

iStar4G: Estendendo a linguagem iStar 2.0 para modelagem de jogos digitais

Sanderson de Paula Barbosa e André Luiz de Castro Leal

Departamento de Computação, UFRRJ
Núcleo de Práticas de Engenharia de Software Aplicadas a Games (NuESGames)
Seropédica, Rio de Janeiro

sandersonbarbosa36@gmail.com, andrecastr@gmail.com

Abstract. Digital games, if considered software products, can and should be built with technical criteria and supported by Software Engineering practices to properly achieve desired qualities. Goal oriented modeling presents itself as an applicable approach to the initial stages of software prospection and development, being able to make explicit stakeholders' intentions and objectives. Applying the iStar 2.0 language to digital game modeling, through a process of reverse engineering, revealed limitations to original elements' usage in actor/agent and resource modeling within the scope of the analyzed games. From such limitations, we propose extending the traditional iStar 2.0 language into *iStar for Games (iStar4G)*, seeking to achieve desired representations to support digital game definition and development.

Keywords: Game Development Process, Game Modeling, , istar, Requirements Engineering, Goal-Oriented Model, GORE.

Resumo. Jogos digitais, se considerados como produtos de software, podem e devem ser construídos com critérios técnicos e suporte da Engenharia de Software para que alcancem as qualidades desejadas. A modelagem orientada a objetivos se apresenta como uma possibilidade aplicável aos estágios iniciais de prospecção do desenvolvimento de softwares, permitindo explicitar intencionalidades e objetivos de *stakeholders* do produto. A aplicação da linguagem *iStar 2.0* na modelagem de jogos, por meio de engenharia reversa, revelou limitações ao uso de seus elementos originais para se modelar atores/agentes e especificação de recursos dos personagens dentro do escopo dos jogos analisados. A partir dessas limitações, propomos a extensão da linguagem *iStar 2.0* tradicional para a linguagem *iStar For Games (iStar4G)*, com o objetivo de atingirmos as representações desejadas para dar suporte à definição e desenvolvimento de jogos digitais.

Palavras-chave: Processo de Desenvolvimento de Jogos, Modelagem de Jogos, istar, Engenharia de Requisitos, Modelagem Orientada a Objetivos.

1 Introdução

Na Engenharia de Software (ES), o processo de construção de software de qualidade consiste em trabalhar diversas etapas, cada uma com sua complexidade específica e possíveis variações de metodologia com uso de procedimentos, técnicas e métodos adequados [1] [2]. Elicitar e especificar necessidades do software em termos de requisitos funcionais, tomado como uma das etapas de construção, requer o uso de linguagens e métodos para se entender e documentar os diversos tipos de requisitos que comporão o software [3]. Como exemplo de linguagem para se modelar software pode-se citar a linguagem iStar [4][5], que apresenta-se como uma abordagem que leva em consideração aspectos sociais e interativos que influenciam a estrutura de requisitos funcionais e não-funcionais [4].

Os jogos digitais podem ser compreendidos como produtos de software se considerarmos algumas intercessões entre eles e os softwares tradicionais, como a necessidade de utilização de recursos computacionais, definições de escopo, prazo e critérios de qualidade [6]. O clássico *Super Mario World*¹, lançado em 1990 para o *Super Nintendo Entertainment System* (SNES), pode ser visto como um título influente no mercado de jogos, e um marco em termos de evolução da jogabilidade, requisito de qualidade, já estabelecida pelos seus predecessores. Por esse motivo, foi o jogo escolhido para esta análise.

O presente trabalho apresenta a modelagem do clássico jogo *Super Mario World* com o uso da linguagem *iStar 2.0* e discute possibilidades iniciais para uma extensão da linguagem, para o que é denominado nesse estudo como *iStar4G* (*iStar for Games*), proposta para a modelagem de jogos digitais. Tendo como base as limitações percebidas na aplicação do *iStar* ao jogo analisado, o *iStar4G* se propõe a dar suporte à definição e especificação dos elementos que compreenderão o produto final. O objetivo central com a aplicação da linguagem na construção de jogos digitais é poder proporcionar uma ferramenta que faça que construtores de jogos digitais a pensem em termos de requisitos e projeto antes de se codificar o jogo.

Para a apresentação do trabalho, o presente documento encontra-se distribuído da seguinte forma: na seção 2 encontra-se a fundamentação teórica; na seção 3 está descrito o método de avaliação; na seção 4, os resultados e discussão; na seção 5, a proposta da linguagem *iStar4G*; e na seção 6, a conclusão.

2 Fundamentação Teórica

O *iStar* é uma linguagem de modelagem orientada a objetivos e agentes [7]. Essa abordagem tem como foco evidenciar interações e compensações de requisitos dentro de um contexto. O *iStar 2.0* foi selecionado para utilização por se tratar de uma evolução do já estabelecido *i** [4] que, por sua vez, tem sido a linguagem predominante na modelagem orientada a objetivos [8].

A proposta dessa abordagem é apresentar os objetivos (*goals*) e atores, com intenções e objetivos distintos, como forma abstrata de expor as características que são interessantes aos *stakeholders* [9][10], além de apresentar os *stakeholders* atuantes no

¹ Super Mario World, 1990, https://www.mariowiki.com/Super_Mario_World.

processo, suas intencionalidades, interações e interdependências. [11][12]. Para Yu [13], essa abordagem deve ser utilizada quando se pretende compreender as correlações entre os envolvidos, de acordo com os diversos atores do sistema; compreender as razões dos processos de decisões ou quando se pretende ilustrar modelos para compreensão dos requisitos.

3 Quanto aos Métodos

O estudo consiste em uma análise qualitativa, com estudo da literatura e com a engenharia reversa de jogos digitais tais como: *Sonic the Hedgehog*², *Orcs Must Die*³, e *Pokémon*⁴, porém com maior foco no uso e nas características do jogo *Super Mario World*, possível a partir do ato de jogar o jogo e analisar os componentes envolvidos. Com base em estudos realizados a respeito dos conceitos e aplicações da modelagem orientada a objetivos, tal como dos elementos associados à linguagem *iStar 2.0*, foi realizada uma análise adicional dos diferentes formatos de interação entre os personagens do jogo *Super Mario World* nos sistemas que compreendem o ambiente virtual.

Foram identificados dois principais formatos distintos dentro do jogo, onde as interações de jogabilidade se mostraram drasticamente diferentes entre si. Estes formatos foram mapeados isoladamente, desconsiderando o papel do jogador como agente responsável por realizar as interações, tal como seus objetivos individuais. Deste modo, a aplicação da linguagem se restringe à proposta que o jogo tem como produto e abstrai os objetivos individuais de diferentes jogadores, que podem interagir com o jogo de maneiras distintas.

Os dois grandes formatos entendidos como parte do sistema são o mapa (Figura 2) e as fases (Figura 3). Por mapa, denominado daqui em diante como *off-stage*, entendem-se os momentos em que o personagem interage com os obstáculos (fases a serem vencidas) estáticos que bloqueiam seu caminho. Neste estado do jogo, as interações são extremamente reduzidas, não existem inimigos e nem estado de falha.



Fig. 1. *Super Mario World*: Mapa do Mundo (*off-stage*)

² Sonic the Hedgehog, 1991, [https://pt.wikipedia.org/wiki/Sonic_the_Hedgehog_\(personagem\)](https://pt.wikipedia.org/wiki/Sonic_the_Hedgehog_(personagem)), Sega Games Co., Ltd.

³ Orcs Must Die, Robot Entertainment, <https://robotentertainment.com/omd3>

⁴ Pokémon, 1996, <https://www.pokemon.com/br/>, Game Freak Inc.

Ao selecionar uma fase, ‘Yoshi’s Island 2’ (Figura 1), inicia-se outro formato de interação, chamado aqui em diante de *on-stage*. Neste estado, as interações são mais dinâmicas. Existem inimigos e outros objetos interativos; estes auxiliam ou atrapalham o personagem a concluir a fase. Na figura 2, podemos identificar alguns destes objetos e personagens: o protagonista Mario, ao centro, e à sua direita um item do tipo casco de tartaruga; e inimigos do tipo *Koopa Troopa* (tartarugas de casco vermelho, na plataforma superior); além dos elementos da interface, na parte superior.



Fig. 2. Super Mario World: ‘Yoshi’s Island 2’ (*on-stage*)

4 Resultados e Discussão

Considerando os formatos estabelecidos na seção 3, foram analisadas as intencionalidades dos personagens autônomos dentro do jogo. Foram obtidos três modelos SR em *iStar 2.0*. Os modelos referem-se, respectivamente, aos dois estados de interação: *off-stage* (Figura 3) e *on-stage* (Figuras 4 e 5).

No estado *off-stage*, o objetivo principal do Herói é ter a ‘princesa em segurança’. Para atingi-lo, tarefas derivativas precisam ser executadas, consistindo em ‘derrotar o chefe’ e ‘abrir o caminho’ até ele, e para isso há necessidade de que fases sejam concluídas. O único processo de decisão delegado ao jogador nesse estado é escolher quais fases tentar.

Extraordinariamente, o modelo foi desenvolvido considerando apenas a intencionalidade do Herói, uma vez que não existe interação entre outros personagens neste formato.

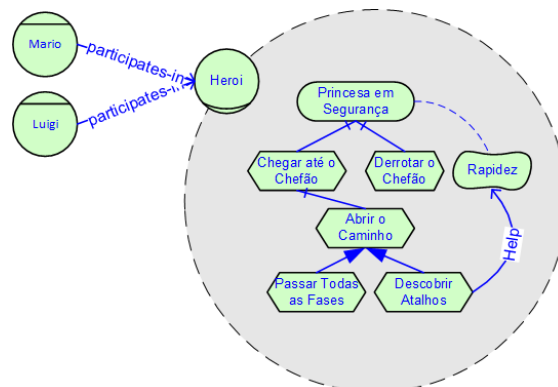


Fig. 3. Modelo SR do Super Mario World: Estado do Mapa (*off-stage*)

Ao optar por entrar em uma determinada fase, o dinamismo das entre ‘Inimigo’ e ‘Herói’ deve ser considerado. Inimigos foram assim considerados por terem a capacidade de interromper o progresso dos heróis. Durante a elaboração do modelo, porém, não foi encontrado um método de associar ambos os papéis diretamente. Segundo a sintaxe de *iStar*, as associações por relações de dependência implicam na colaboração entre os agentes envolvidos [7]. Por este motivo, foram feitos dois modelos SR, referentes ao estado *on-stage*: um do Inimigo e um do Herói, e a repetição da qualidade ‘longevidade’ nos dois modelos é usada para representar que o objetivo do Inimigo impede que essa qualidade, associada ao objetivo do Herói, seja atingida.

Entende-se por ‘longevidade’ neste modelo a qualidade de se cumprir o objetivo especificado, de concluir a fase, sem perder vidas extras (*1-ups*)⁵. ‘Derrotar inimigos’ é uma tarefa opcional, que se executada auxilia na busca pela qualidade ‘longevidade’, custando, porém, tempo. Adicionalmente, mecânicas de interação, como o ‘Pulo’, foram consideradas como recursos, que permitem que o Herói cumpra suas tarefas.

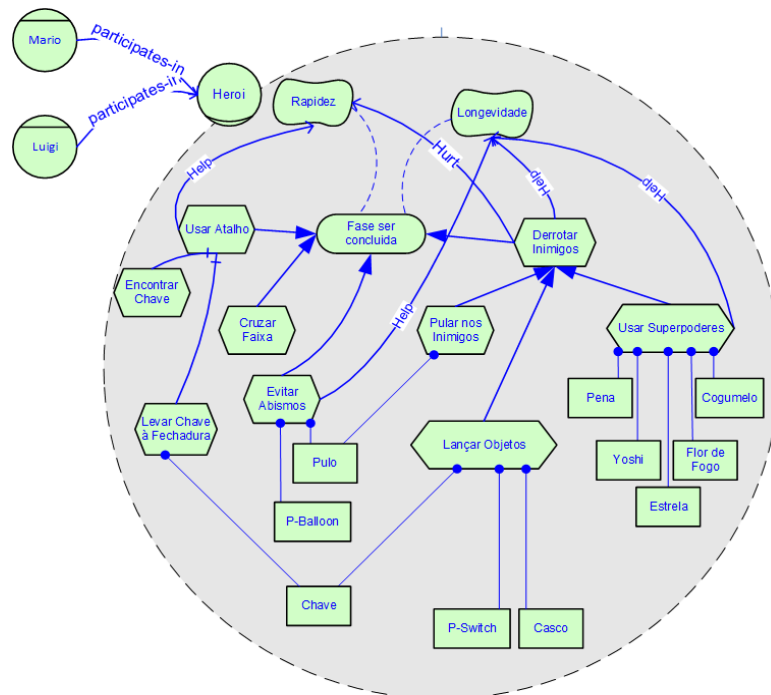


Fig. 4. Modelo SR do SMW: Estado das Fases (*on-stage*): Heróis

⁵ *1-ups* referem-se às vidas extras no Super Mario World. São itens específicos que, quando adquiridos, dão ao personagem uma chance a mais de entrar em uma fase.

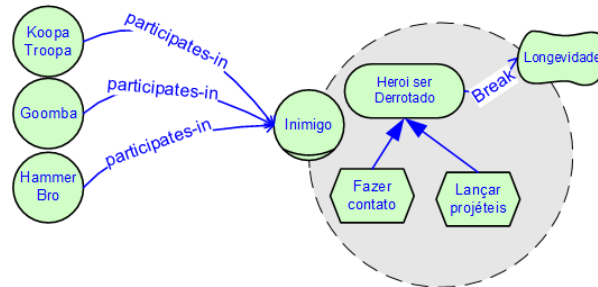


Fig. 5. Modelo SR do SMW: Estado das Fases (*on-stage*): Inimigos

5 Adaptação da Linguagem: o *iStar4G*

Considerando que o *iStar 2.0* não foi elaborado para o contexto de jogos digitais, a tentativa de aplicá-lo nesse estudo evidencia algumas limitações. Houve a necessidade da inclusão de atores sem relações entre si, separando-os em modelos diferentes; e a utilização dos recursos não evidencia com clareza as significativas diferenças que eles têm entre si em termos de comportamento e influência no mundo do jogo.

Os estudos realizados irão evoluir para a proposição de adaptações sintáticas e criação de novos elementos para que se possa representar com maior clareza interações e especificações que precisaram ser explicitadas textualmente. O estudo aqui presente, apesar de incipiente, já é um primeiro esforço dos pesquisadores para propor uma linguagem orientada a objetivos adaptada e com suporte em *iStar 2.0* para ser utilizada na modelagem de requisitos de jogos, o *iStar4G*.

5.1 Novos elementos no *iStar4G*

No cenário do jogo, há uma diferenciação considerável entre recursos mecânicos de interação entre personagem e mundo; objetos manuseáveis pelo personagem; objetos não manuseáveis que modificam os métodos de interação com o mundo, dando ou restringindo capacidades; e objetos, manuseáveis ou não, que modificam o comportamento ou configuração do cenário em que o personagem se encontra.

Com base nesses quatro tipos de recurso diferentes e suas funções dentro do sistema, propomos especificá-los visualmente dentro do recurso original. Na figura 6 estão os ícones de especificação propostos (lado esquerdo) e sua aplicação (lado direito).

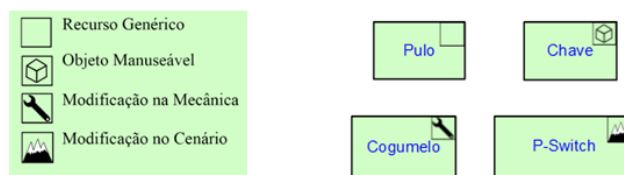


Fig. 6. Ícone e descrição dos novos elementos.

Adicionalmente, entendemos que a representação de relações de antagonismo entre diferentes atores pode ser uma limitação significativa à modelagem em jogos, onde

relações de inimizade são comuns. Por isso, propomos novos links de qualificação, baseados nas pré-existentes *Help*, *Hurt*, *Make* e *Break*. Com isso, passa a ser possível uma relação de antagonismo entre diferentes atores. Na figura 7, representamos visualmente essas novas relações e sua equivalência, em intensidade, às já existentes.



Fig. 7. Proposta de novas qualificações, entre atores diferentes.

5.2 Modelo *Strategic Rationale* em *iStar4G*

Na Figura 8, podemos encontrar um modelo SR alternativo ao das figuras 4 e 5, levando em consideração as alterações propostas na sessão 5.1. Nesta nova linguagem, torna-se possível incluir ambos os atores na mesma figura, adotando as novas qualificações; além de especificar com maior clareza os diferentes recursos do jogo.

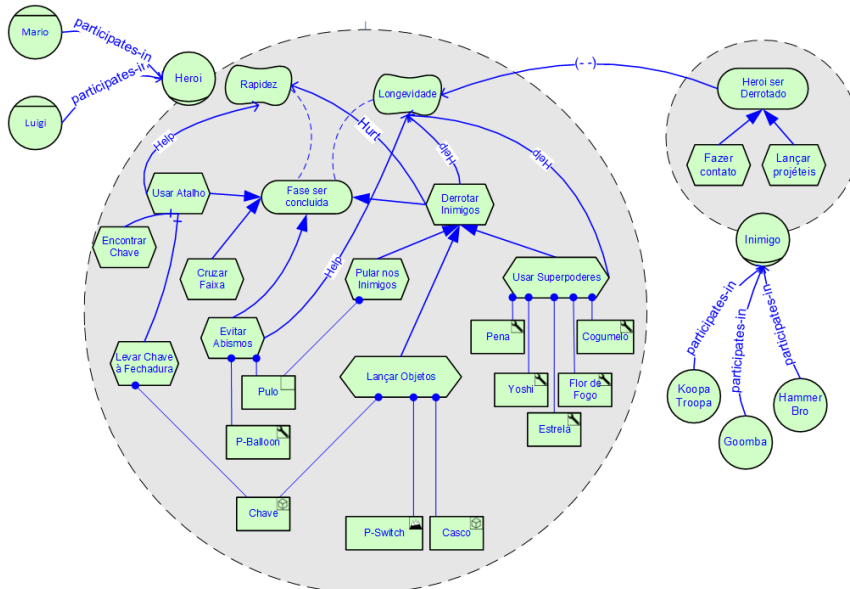


Fig. 8. Modelo SR em *iStar4G*: *Super Mario World (on-stage)*

6 Conclusão

Utilizando a linguagem *iStar 2.0*, verificamos que é possível modelar aspectos da interação entre personagens de jogos digitais em alto nível de abstração, referente aos estágios iniciais de sua prospecção e construção. Acreditamos que a utilização desta abordagem pode dar suporte ao desenvolvimento do já utilizado *Game Design Document* na especificação de requisitos funcionais e não funcionais para jogos digitais.

Para a posteridade, consideramos: uma melhor estruturação do *iStar4G* tendo como base o estado da arte na pesquisa sobre desenvolvimento de jogos e extensões do *iStar*; validar do uso do *iStar4G* por meio de novos casos e a elaboração de mais ar-

quétipos e relações a partir das limitações encontradas, como no estudo aqui presente; aplicar o *iStar4G* na modelagem de um jogo ainda não existente, averiguando também sua integração com processos já estabelecidos de desenvolvimento de jogos; ou o desenvolvimento de ferramentas que permitam a modelagem em e adoção de *iStar4G* pela comunidade de engenheiros e desenvolvedores.

Agradecimentos

Os pesquisadores envolvidos no estudo agradecem à Universidade Federal Rural do Rio de Janeiro pela bolsa PIBIC 2019.

Referências

1. PRESSMAN, R. S. (2016). Engenharia de software. São Paulo: Makron Books.
2. SOMMERVILLE, I. (2019). Software Engineering, Global Edition. Addison Wesley, 2019.
3. BOURQUE, P. and FAIRLEY, R.E., eds. (2014), Guide to the Software Engineering Body of Knowledge, Version 3.0, IEEE Computer Society, 2014; www.swebok.org. – Acessado em 30 de Setembro de 2019.
4. YU, E. S. K. (1995). Modelling Strategic Relationships For Process Reengineering. Ph. D. dissertation. Dept. of Computer Science, University of Toronto, 1995.
5. iStar Wiki (2019). <Disponível em: <http://istar.rwth-aachen.de/tiki-index.php>> <Acessado em: 30 de setembro de 2019>.
6. RUCKER, R. (2002) Software Engineering and Computer Games, Addison-Wesley, New Jersey.
7. DALPIAZ, F.; FRANCH, X.; and HORKOFF, J.. (2016). iStar 2.0 Language Guide.
8. HORKOFF, J. et al., "Goal-Oriented Requirements Engineering: A Systematic Literature Map," 2016 IEEE 24th International Requirements Engineering Conference (RE), Beijing, 2016, pp. 106-115
9. ALI, R., DALPIAZ, F. and GIORGINI, P. (2010). A goal-based framework for contextual re-quirements modeling and analysis. Requirements Engineering, 15(4), pp.439-458.
10. YU E., MYLOPOLOUS J, (1998) Why goal-oriented requirements engineering. In: Du-bois E, Opdahl AL, Pohl K (eds) Proceedings of the 4th international workshop on re-quirements engineering: foundation for software quality (RESFQ 1998). Presses Universi-taires de Namur, Namur
11. LAPOUCHNIAN, A. (2005). Goal-oriented requirements engineering: An overview of the cur-rent research. University Of Toronto, 32. Retrieved from <http://www.cs.toronto.edu/~alexei/pub/Lapouchnian-Depth.pdf>
12. FRANCH, X., LÓPEZ, L., CARES, C., & COLOMER, D. (2016). The i* Framework for Goal-Oriented Modeling. Domain-Specific Conceptual Modeling. Springer, Cham. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-39417-6_22
13. YU, E. (2002). Agent-Oriented Modelling: Software Versus World, In: Proceedings of the Agent-Oriented Software Engineering, Edited by Wooldridge, M., Weiss, G. and Cianc-rini, P., LNAI, 2222, Springer-Verlag, p. 206 – 225