

Proposta de um Processo de Desenvolvimento de Jogos Educativos

Carla Silva, André Calisto, David Barbosa

Departamento de Ciências Exatas, Centro de Ciências Aplicadas e Educação –
Universidade Federal da Paraíba (UFPB)
58.297-000 – Rio Tinto – PB – Brazil

{carla, andre.calisto, david.barbosa}@dce.ufpb.br

***Abstract.** This paper proposes an adaptation of a software agent based process to develop intelligent games for education.*

***Resumo.** Este artigo propõe a adaptação de um processo baseado em agentes de software para o desenvolvimento de jogos inteligentes para a educação.*

1. Introdução

Os jogos educativos (do inglês, Game Based Learning ou GBL) constituem um ramo dos chamados jogos sérios [GBL 2007 apud Pádua 2008] que têm como objetivo instruir e educar os jogadores sobre algum tema ou conceito, além de proporcionar entretenimento e diversão. De acordo com Amory [Amory, 2001], os jogos educativos disponibilizam uma forma onde o aprendiz pode estar imerso em micro mundos construtivistas. Os jogos educativos computacionais são atividades inovadoras, onde as características do processo de ensino e aprendizagem apoiado no computador e nas estratégias do jogo são integradas a fim de alcançar o objetivo educacional procurado [Moratori, 2003 apud Pádua, 2008]. Para tanto, este tipo de sistema deverá manipular grandes informações, representadas de maneiras diversas, através de imagens, textos, sons, filmes etc. Um jogo educativo, na maioria das vezes, faz uma representação funcional da realidade, podendo ser um modelo computacional simplificado misturando características de um jogo – competição, cooperação, regras etc. – com aquelas de um simulador, que é a incorporação de características críticas da realidade [Silva, A. et al., 2007] reguladas por regras e restrições.

À medida que os jogos se tornam mais complexos, os desenvolvedores de jogos estão começando a reconhecer a necessidade de se construir personagens mais inteligentes e, portanto, eles são obrigados a colocar maior ênfase no desenvolvimento da Inteligência Artificial (IA) em seus jogos [Laird & van Lent, 1999]. Um dos objetivos fundamentais da IA, é entender e implementar sistemas inteligentes que apresentem todas as capacidades de um ser humano [Wooldridge, 2002].

Vários processos de desenvolvimento de jogos foram propostos tendo como base modelos de processo de software bem conhecidos, tais como o modelo em cascata, o processo unificado e processos ágeis [Rodrigues, H. et al., 2010]. Este artigo propõe um processo de desenvolvimento de jogos somente sob a forma de ambientes virtuais com regras e restrições específicas que possam ser populados por agentes inteligentes [Silva, F., 2005]. Isto exclui jogos de tabuleiro e semelhantes. Como o ambiente virtual onde

um jogo acontece pode ser tão complexo quanto se queira, implementar o comportamento inteligente nos agentes que vivem nesse ambiente é que motiva a adaptação de uma metodologia de desenvolvimento de sistemas multi-agentes para o desenvolvimento de jogos interativos. Em particular, a metodologia de desenvolvimento alvo deste trabalho é o Tropos [Castro et al., 2002] - um framework para o desenvolvimento de software orientado a agentes para ambientes dinâmicos e distribuídos. Nosso objetivo é adaptar o framework Tropos para que ele guie o desenvolvimento de jogos interativos e inteligentes para educação. Um jogo inteligente auxiliará na educação sem deixar de lado o entretenimento proporcionado pelos jogos interativos. Portanto, é preciso definir um processo para o desenvolvimento de jogos educativos que se beneficie da capacidade que os sistemas multi-agentes têm de simular computacionalmente um ambiente.

2. Tropos4EG (Tropos for Education Games)

Nesta seção encontra-se a adaptação do framework Tropos, originalmente definido para o desenvolvimento de sistemas de informação, para permitir a especificação e o desenvolvimento de jogos inteligentes para educação. Para tanto, adaptamos o processo do Tropos, especialmente para as fases de requisitos e arquitetura.

A Figura 1 apresenta uma primeira versão desta adaptação. Neste estágio do trabalho, o processo possui apenas duas atividades, que se preocupam em elicitar e especificar as características iniciais do jogo educativo.

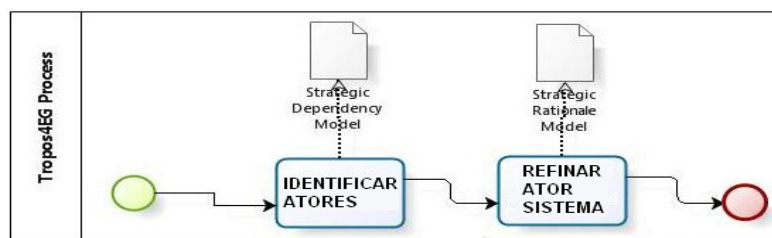


Figura 1. Processo do Tropos4EG

Identificar Atores: Na primeira atividade, utiliza-se o framework *i** para identificar os atores estratégicos da aplicação, incluindo o ator sistema, bem como as dependências estratégicas entre estes atores. Observamos que dois atores sempre estarão presentes no modelo, o ator Aluno e o ator Tutor, pois eles serão os stakeholders que definirão as características que o sistema deverá apresentar.

O Tropos adota dois modelos: o modelo SD (*Strategic Dependency* – Dependência Estratégica) e o modelo SR (*Strategic Rationale* – Raciocínio Estratégico). O modelo SD apresenta os atores e suas dependências, enquanto que o modelo SR apresenta os detalhes internos dos atores.

No modelo SD um ator pode depender de outro para satisfazer um objetivo (o ator Estudante depende do ator Jogo para alcançar o objetivo Encontrar Amigos), para executar uma tarefa, fornecer um recurso ou satisfazer um softgoal (o ator Tutor depende do ator Jogo para alcançar o softgoal Usabilidade). Softgoals estão relacionados a requisitos não-funcionais, enquanto objetivos, tarefas e recursos estão relacionados com funcionalidades do software [Castro et al., 2002]. O ator que depende do outro é

chamado de Dependere enquanto que o outro ator que recebe a requisição é chamado de Dependee. No framework i*, um ator pode representar um usuário do sistema, o próprio sistema em desenvolvimento ou um sistema pré-existente.

Na Figura 2 aparece o detalhamento do ator que representa o jogo que será desenvolvido, mostrando seus elementos internos e suas relações com as dependências externas. Isto corresponde à segunda atividade do processo.

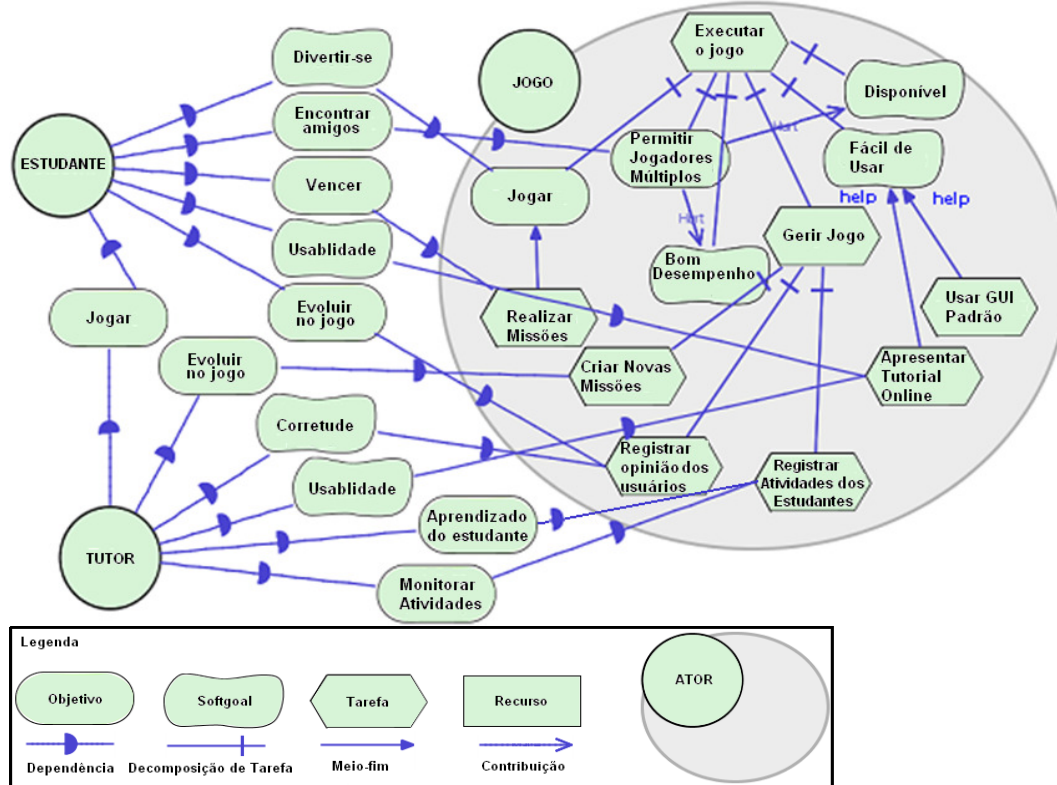


Figura 2. Modelo Inicial de Requisitos do Jogo Educativo

Refinar Ator Sistema: No modelo SR, onde os atores são detalhados, aparecem novos tipos de relacionamentos, tais como meio-fim, decomposição-tarefa e contribuição. Alguns requisitos não-funcionais de jogos interativos e softwares educativos, tais como disponibilidade, desempenho, usabilidade, deverão estar sempre presentes no modelo. Um relacionamento de decomposição-tarefa existe entre uma tarefa e suas partes, mostrando como uma tarefa é executada. Na Figura 1, o ator Game apresenta a tarefa raiz Executar o Jogo, que é refinada, através do relacionamento de decomposição-de-tarefa, no softgoal Disponível, Bom desempenho e Fácil de Usar, dos objetivos Jogar e Permitir Múltiplos Jogadores e da tarefa Gerir Jogo. Esta última tarefa é refinada nas tarefas Criar Novas Missões, Registrar Opinião dos Usuários e Registrar Atividades dos Estudantes.

Um relacionamento meio-fim descreve uma ligação de um elemento meio para alcançar um objetivo (fim). Um meio é uma alternativa para realizar um fim (normalmente um elemento do tipo objetivo) mostrando como este objetivo pode ser atingido. Por exemplo, no ator Jogo (Figura 2), o objetivo Jogar é refinado, através do relacionamento meio-fim, na tarefa Realizar Missões.

O relacionamento do tipo contribuição descreve uma contribuição de um meio (tarefa ou objetivo) para a realização de um fim (softgoal). Este tipo de relacionamento fornece um raciocínio qualitativo usando um esquema de valores para representar a contribuição (e.g., Help e Hurt significam Ajuda e Prejudica, respectivamente) [Castro et al., 2002]. Por exemplo, na Figura 2, a tarefa Usar GUI Padrão contribui positivamente (Help ou Ajuda) para a satisfação do softgoal Fácil de Usar.

3. Conclusão e Trabalhos Futuros

Este artigo apresentou uma proposta inicial de um processo de desenvolvimento de jogos educativos baseado no framework Tropos. Uma especificação inicial das características de um jogo educativo foi apresentada. Espera-se que muitas destas características sejam compartilhadas por jogos educativos, dando precedentes para construirmos jogos educativos como uma Linha de Produto de Software [Clements and Northrop, 2002]. Isto só poderá ser confirmado após a especificação de vários jogos educativos para disciplinas distintas.

Agradecimentos

Este trabalho foi financiado pelo CNPq, processo nº 501646/2009-4.

Referências

- Castro, J., et al. Towards Requirements-Driven Information Systems Engineering: The Tropos Project. *Information Systems Journal*, Elsevier, 27(6): 365-389, 2002.
- Clements, P., Northrop, L. *Software Product Lines: Practices and Patterns*. Addison-Wesley, 2002.
- Game Based Learning. Texto explicativo sobre game based learning (gbl). 2007. URL: <http://www.seriousgames.org>. Data do último acesso: 07/10/2010.
- Laird, J. and van Lent, M. Intelligent agents in computer games. In *Proceedings of the National Conference on Artificial Intelligence*, pages 929–930. AAAI Press, 1999.
- Moratori, P. B. Por que Utilizar Jogos Educativos no Processo de Ensino Aprendizagem? Trabalho de Conclusão da Disciplina Introdução a Informática na Educação. Universidade Federal do Rio de Janeiro, 2003.
- Pádua, V. C. Ambiente de suporte a jogos WEB voltado para a área de ensino a Distância. Dissertação de Mestrado. CIn, Universidade Federal de Pernambuco, 2008.
- Rodrigues, H.F., et al. Definição e Aplicação de um Modelo de Processo para o Desenvolvimento de Serious Games na Área de Saúde. In: *Proceedings of the CSBC'10 - Workshop de Informática Médica*, p. 1532-1541. Belo Horizonte, 2010.
- Silva, F. *Agentes Inteligentes em Jogos de Computador*. Monografia. Instituto de matemática e estatística. Universidade de são Paulo, 2005.
- Silva, A., Passerino, L. A Fazenda - Software Educativo para a Educação Ambiental. *RENOTE. Revista Novas Tecnologias na Educação*, v. 5, p. 1-12, 2007.
- Wooldridge, M. *Introduction to Multiagent Systems*, John Wiley and Sons, New York, 2002.