Criação de Jogos Didáticos de Perguntas e Respostas por meio de Modelagem em Redes de Petri Coloridas Hierárquicas

Vanessa Viana S. Carvalho¹, Carlos Hairon R. Gonçalves², Joel Cruz Soares¹, Felipe Mota Barreto¹, José Marques Soares¹, Giovanni C. Barroso¹

¹Universidade Federal do Ceará (UFC)

²Instituto Federal do Ceará (IFCE)

vanessaviana@great.ufc.br, hairon@ifce.edu.br, joelcruz007@gmail.com
felipemota0301@gmail.com, marques@ufc.br, gcb@fisica.ufc.br

Abstract. This paper presents a method to create question-and-answer games only using Hierarchical Coloured Petri Nets. The rules, properties and structures are specified in the modelling that are automatically instantiated from the interpretation of the model. The created model can be formally analysed and all the inconsistencies and mistakes can be identified. To validate this method, it is shown the CPN Quiz tool and a didactic example of a question-and-answer game made by this tool.

Resumo. Este trabalho apresenta um método para a criação de jogos de perguntas e respostas exclusivamente por meio de modelagem em de Redes de Petri Colorias Hierárquicas. As regras, propriedades e estruturas do jogo são especificadas na modelagem e automaticamente instanciadas a partir da interpretação do modelo. O modelo criado pode ser analisado formalmente e todas as inconsistências e erros podem ser identificados. Para validar o método, é apresentada a ferramenta CPN Quiz e um exemplo didático de um jogo de perguntas e respostas criado por essa ferramenta.

1. Introdução

Os jogos educativos vêm sendo progressivamente incluídos no processo da educação [Prensky 2005], [Hwang and Wu 2012]. Vários são os exemplos de jogos dessa natureza [Barbosa et al. 2008], [Grübel and Bez 2006], [Mann et al. 2002], [Dalla Vecchia et al. 2014] e [Pietruchinski et al. 2011]. O jogo *Refraction*¹ é um exemplo de um jogo educativo que foi financiado pela *Bill & Melinda Gates Foundation*². Porém, as dificuldades existentes na criação de um jogo eletrônico são as mesmas existentes na criação de qualquer outro software. Tais problemáticas vêm sendo debatidas no contexto da Engenharia de Software desde o início da década de 1970 [Petrillo et al. 2009]. Entre as dificuldades encontradas nessa área, de maneira geral, percebe-se que a rápida obsolescência de ferramentas e tecnologias de desenvolvimento. Uma maneira de minimizar tal problema é o reuso de estruturas que permitam ao desenvolvedor o mínimo ou mesmo nenhuma programação, quando possível.

¹http://centerforgamescience.org/portfolio/refraction/

²http://www.gatesfoundation.org/

[Resnick et al. 2009] e [Torrente et al. 2010] são exemplos de ferramentas que permitem o desenvolvimento de jogos com as características mencionadas acima.

Neste trabalho é apresentada uma ferramenta que viabiliza a concepção e construção de jogos de perguntas e respostas, a partir de modelos em Redes de Petri Coloridas (RPC). Com a possibilidade de modelar hierarquicamente a RPC, o designer do jogo pode criar jogos de Perguntas e Respostas em que o usuário vái avançando no jogo de acordo com o nível modelado. O arquivo de codificação do modelo em RPC é processado e gera uma instância do jogo, não sendo necessária qualquer codificação adicional após a construção do modelo. As localidades, caminhos e regras estabelecidas para o jogo derivam das características da RPC, que é modelada com o uso da ferramenta CPN Tools³.

Este trabalho está estruturado da seguinte maneira: na Seção 2 são introduzidos os conceitos básicos, tais como Redes de Petri Coloridas e Hierárquicas e os trabalhos relacionados. Na Seção 3 é apresentado o as regras de modelagem. Na Seção 4 são introduzidos conceitos de RP hierárquica para modelagem de multiníveis. Na Seção 5 são apresentadas as considerações finais.

2. Uso de Redes de Petri para apoio à concepção de jogos eletrônicos

As Redes de Petri (RP) constituem uma ferramenta matemática para modelagem de sistemas, possuindo uma representação gráfica formada por arcos e nós [Murata 1989]. Graficamente, a estrutura básica das Redes de Petri (RP) são os lugares, arcos e transições, que são representados por elipses, arcos direcionais e retângulos, respectivamente. A dinâmica de uma RP se dá pelo disparo de uma transição que remove fichas dos lugares de entrada e deposita fichas nos lugares de saída.

As RPs vêm sendo empregadas como instrumento de suporte ao desenvolvimento de jogos eletrônicos em suas múltiplas características. Alguns exemplos são apresentados em [Lee and Cho 2011], [Lee and Cho 2012], [Araújo and Roque 2009] e [Santos and Roque 2010]. Porém, eles não permitem representações de jogos multinível por meio de Redes de Petri hierárquicas. Além disso, não foram encontrados trabalhos que permitam a geração de jogos de maneira automatizada a partir de modelos concebidos em RP.

Dentre as diversas ferramentas que permitem a modelagem por Redes de Petri⁴, foi adotada neste trabalho a ferramenta CPN Tools. A integração dessa ferramenta permite a modelagem da estrutura e do comportamento do jogo, sendo possível realizar simulações e a análise formal da rede.

3. Regras para modelagem de jogos

As personagens do jogo são definidas pela configuração das fichas, que são colocadas nos lugares e representam os *sprites* (personagens em movimento). Cada ficha representa um único *sprite*. Os *sprites* são associados a uma representação gráfica através de identificadores numéricos configurados nas fichas. A imagem *A* da Figura 1 apresenta um exemplo em que uma personagem é representada pelo valor inteiro 1, identificado pela marcação

³Ferramenta disponível em http://cpntools.org/

⁴Lista disponível em http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html

1'1 (significando uma ficha com o valor 1). A personagem pode se deslocar entre os caminhos (abstração feita para os lugares neste exemplo de jogo) de acordo com as regras modeladas na RPC. Por exemplo, a RPC da imagem *A* da Figura 1 especificam que a personagem pode percorrer os caminhos nos sentidos das setas. A imagem *C* da Figura 1 apresenta a interface do jogo gerado a partir da RPC da imagem *A* da Figura 1. O jogo modelado define a estrutura do jogo e as regras para o deslocamento de personagens por vários lugares.

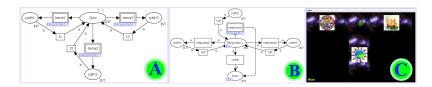


Figura 1. Na imagem *A*, exemplo de uma Rede de Petri Colorida Hierárquica. Na imagem *B*, sub-rede *Pergunta11*, que processa a transição de substituição *tema1* da rede da imagem *A*. Na imagem *C*, é apresentado o jogo gerado a partir da rede da imagem *A*.

As hierarquias em uma RPC são especificadas por meio de transições de substituição. Estas são representadas por retângulos de borda dupla, como a transição identificada por *tema1*, *tema2* e *tema3* da imagem *A* da Figura 1. Transições de substituição representam um nível de abstração para uma sub-rede. No exemplo apresentado, a transição *tema1* da imagem *A* da Figura 1 é a abstração da sub-rede apresentada na imagem *B* da Figura 1.

É adotado as transições de substituição para configurar múltiplos níveis no jogo. Dessa maneira, ao deslocar um *sprite* para uma transição de substituição, a personagem é transferida para outro domínio, em que a estrutura de deslocamento é constituída por novos lugares e transições que são associados por regras distintas àquelas do nível anterior. Do ponto de vista da interface gráfica, o deslocamento para uma transição de substituição implica na mudança da fase do jogo, com consequente modificação da imagem do cenário.

4. Projeto de Jogos Multiníveis de Perguntas e Respostas

A possibilidade de usar uma RPC hierárquica pode ajudar o projetista com mais opções para realizar a modelagem do jogo. A hierarquia pode ser modelada de forma que pode representar várias fases ou até uma passagem para o cumprimento de uma tarefa extra.

Para a construção do jogo, cada pergunta e cada resposta devem corresponder a uma imagem. Um exemplo pode ser visto na Figura 2, onde para a transição de substituição *tema2* da imagem *A* da Figura 1, foi utilizada a imagem *A* da Figura 2, e para o lugar de entrada *Pergunta1.1* foi utilizada a imagem *B* da Figura 2. Essas imagens serão interpretadas pela ferramenta como especificado no modelo.

Sendo as sub-redes representadas em uma rede de nível superior por transições de substituição, a personagem pode se deslocar pelos múltiplos níveis do jogo através da utilização de portais específicos. No exemplo de jogo apresentado na imagem C da Figura 1, esses portais são representados por uma imagem de um tema. O jogo se desenvolve de maneira que, a partir de interações do usuário com a interface gráfica, o *sprite* se movimenta entre os diversos lugares existentes na rede. Além disso, ao fazer o *sprite* colidir

com a representação de um portal, migra-se para um nível diferente do jogo, momento em que é apresentado ao usuário o *layout* correspondente àquele nível. Regras podem ser estabelecidas via modelagem de maneira que apenas os *sprites* que tenham alcançado determinadas propriedades possam mudar de nível.

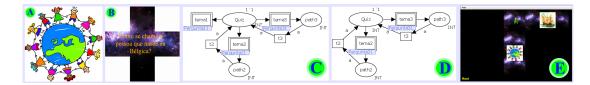


Figura 2. A imagem A substitui uma transição de substituição (tema2). A imagem B substitui um lugar de entrada (Pergunta2.1). Na imagem C, rede da imagem A da Figura 1 sem os lugares e as transições de retorno. Na imagem D, rede da imagem C sem a transição tema1 e o lugar path1. Na imagem E, CPN Quiz apresentando a geração do jogo relativo à rede.

O modelo em que foram definidas as regras e a estrutura do jogo pode ser modificado exclusivamente a partir do CPN Tools. O novo modelo pode, então, ser carregado e interpretado pelo *CPN Quiz*. Para exemplificar esse procedimento, ao retirar-se a transição de substituição *tema1*, a transição *t1* e o lugar com o nome *path1* da imagem *C* da rede da Figura 2, obtém-se a rede apresentada na imagem *D* da Figura 2. Em seguida, carregando-se o modelo modificado na ferramenta *CPN Quiz*, a imagem que representa a transição e a imagem que representa o lugar retirado do modelo não serão mais apresentadas, como é apresentado na imagem *E* da Figura 2, sem que seja necessária uma linha sequer de programação. Após realizar a modelagem ou fazer qualquer alteração em modelos existentes, devem-se reexecutar os procedimentos de análises e simulações por meio da ferramenta CPN Tools para validação do novo modelo.

5. Conclusões

Neste trabalho é apresentado um método que permite a construção de jogos de perguntas e respostas com múltiplos níveis exclusivamente por modelagem em RPC. Modelos permitem que o projetista faça testes e simulações antes da implementação efetiva do software. No *CPN Quiz*, não é necessária qualquer atividade de programação adicional após a construção e análise do modelo. Dessa maneira, esse método de construção permite que pessoas com pouco conhecimento de programação concebam seus próprios jogos. Com o conjunto de ferramentas disponibilizadas nesse trabalho, é possível criar facilmente um leque de jogos educacionais de concepção simples com baixo custo e alta produtividade.

Agradecimentos

Ao CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico, instituição científica da maior respeitabilidade no meio acadêmico brasileiro, o nosso agradecimento pela parceria na realização deste artigo.

Referências

Araújo, M. and Roque, L. (2009). Modeling games with petri nets. *Breaking New Ground: Innovation in Games, Play, Practice and Theory. DIGRA2009. Londres, Royaume Uni.*

- Barbosa, R. C., Tavares, R., Santos, J. N. d., Rodrigues, G. L., and Andrade, M. (2008). O jogo educacional como recurso digital ea aprendizagem significativa de gramática. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 1, pages 491–500.
- Dalla Vecchia, R., Maltempi, M. V., and Weingarten, T. (2014). A construção de jogos eletrônicos ea modelagem matemática na realidade do mundo cibernético. *EDUCAÇÃO MATEMÁTICA EM REVISTA-RS*, 2(14).
- Grübel, J. M. and Bez, M. R. (2006). Jogos educativos. RENOTE, 4(2).
- Hwang, G.-J. and Wu, P.-H. (2012). Advancements and trends in digital game-based learning research: a review of publications in selected journals from 2001 to 2010. *British Journal of Educational Technology*, 43(1):E6–E10.
- Jensen, K. and Kristensen, L. M. (2009). *Coloured Petri nets: modelling and validation of concurrent systems*. Springer.
- Lee, Y.-S. and Cho, S.-B. (2011). Context-aware petri net for dynamic procedural content generation in role-playing game. *Computational Intelligence Magazine, IEEE*, 6(2):16–25.
- Lee, Y.-S. and Cho, S.-B. (2012). Dynamic quest plot generation using petri net planning. In *Proceedings of the Workshop at SIGGRAPH Asia*, pages 47–52. ACM.
- Mann, B. D., Eidelson, B. M., Fukuchi, S. G., Nissman, S. A., Robertson, S., and Jardines, L. (2002). The development of an interactive game-based tool for learning surgical management algorithms via computer. *The American Journal of Surgery*, 183(3):305–308.
- Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.
- Petrillo, F., Pimenta, M., Trindade, F., and Dietrich, C. (2009). What went wrong? a survey of problems in game development. *Computers in Entertainment (CIE)*, 7(1):13.
- Pietruchinski, M. H., Coelho Neto, J., Malucelli, A., and Reinehr, S. (2011). Os jogos educativos no contexto do sbie: uma revisão sistemática de literatura. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 1.
- Prensky, M. (2005). Computer games and learning: Digital game-based learning. *Hand-book of computer game studies*, 18:97–122.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., et al. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11):60–67.
- Santos, S. D. and Roque, L. G. (2010). Ensaio de reescrita de comportamentos em videojogos com base no ajuste e computação de modelos de petri net. *IX SBGames*.
- Torrente, J., Del Blanco, Á., Marchiori, E. J., Moreno-Ger, P., and Fernández-Manjón, B. (2010). < e-adventure>: Introducing educational games in the learning process. In *Education Engineering (EDUCON)*, 2010 IEEE, pages 1121–1126. IEEE.