

Simulação Baseada em Atores para no Ensino de Arquitetura de Computadores

Alisson V. Brito

Universidade Federal da Paraíba

Centro de Ciências Aplicadas e Educação (CCAIE)

Departamento de Ciências Exatas

alissonbrito@cae.ufpb.br

Resumo

Este trabalho apresenta um projeto em desenvolvimento de uma ferramenta de simulação de sistemas concorrentes baseada em atores, que permite os estudantes criarem seus próprios modelos de sistemas de uma forma interativa e lúdica. Como um estudo de caso, atores de uma CPU estão sendo desenvolvidos, permitindo os estudantes projetarem e construir seus próprios modelos de processadores durante o curso de arquitetura de computadores.

1. Introdução

Muitas técnicas de ensino de arquitetura e organização de computadores utilizam de simuladores de arquiteturas reais ou hipotéticas [1]. Estas técnicas, porém, não permitem uma abordagem construtivista onde os alunos possam criar suas próprias arquiteturas.

A simulação de sistemas concorrentes baseada em atores possui um grande potencial educacional ainda pouco explorado. Este trabalho apresenta uma ferramenta de simulação que permite o aluno criar seus próprios modelos de sistemas concorrentes de forma interativa e lúdica. Os primeiros passos no desenvolvimento desta ferramenta foram desenvolvidos e são descritos neste trabalho através de uma metodologia para criação de atores e materiais didáticos.

Aqui é apresentada uma ferramenta de simulação que permite o aluno criar seus próprios ambientes, simular praticamente qualquer sistema concorrente e estimular a formação de novos atores através da composição de atores já existentes, tendo como principais características o incentivo à criatividade e ao entretenimento.

1.1. Simulação por Modelos de Atores e o Ptolemy

Um Modelo de Atores é um modelo matemático de computação concorrente que trata “atores” como sendo os elementos primitivos da computação concorrente digital. Em resposta a uma mensagem recebida por um ator, ele toma uma decisão local, cria mais atores, envia mais mensagens e determina como responder à próxima mensagem recebida [2]. O Modelo de Atores foi originado em 1973 no artigo [3], relatando sua importância como base teórica para o entendimento teórico de concorrência e como ferramenta para a implementação prática de sistemas concorrentes.

O Modelo de Atores se qualifica como uma ferramenta adequada para a modelagem de ambientes estudados pelas ciências básicas vistas no ensino médio e fundamental, isso por apresentar duas características principais, a concorrência e a orientação a mensagens [4], [6].

A concorrência implica que os atores serão executados simultaneamente, já a orientação a mensagens significa que os atores serão configurados para receber mensagens, reagir a essas mensagens e gerar novas mensagens na saída para outros atores. Por exemplo, no ensino da Matemática, uma equação de duas variáveis pode ser um ator, que recebe como entrada uma mensagem com o valor de uma das variáveis, gerando na saída uma mensagem com o valor da outra variável, já na Biologia, um vírus pode ser um ator, recebendo como entrada uma mensagem contendo proteínas, reagindo a elas e produzindo mensagens com novas proteínas, que podem ser absorvidas por outros vírus próximos.

Uma vez que os atores tenham sido modelados, ambientes podem ser organizados de forma que vários atores interajam trocando mensagens, reagindo e evoluindo de acordo com os estímulos gerados pelos próprios atores do mesmo ambiente de forma

independente. Tudo isso pode ser exibido aos alunos passo-a-passo, com animações, uso de diferentes cores, desenhando gráficos e reproduzindo sons, gerando recursos mais atrativos. Outro forte dessa abordagem é que o aluno não será elemento passivo na execução das simulações, a tarefa passada pelo professor pode ser a de criar um novo ambiente de simulação, escolhendo os atores e combinando-os para alcançar a especificação dada pelo professor.

O paradigma de modelagem de sistemas através de atores possui um grande potencial ainda pouco explorado de ferramenta educacional. Uma vez que os atores tenham sido bem modelados e desenvolvidos, professores podem gerar ambientes multidisciplinares que mostrem aos alunos de forma interativa (muitas vezes com recursos de animação 3D e som) como os atores se relacionam e geram efeitos no ambiente e nos atores vizinhos.

1.2. O projeto Ptolemy

O projeto Ptolemy, criado e mantido na Universidade de Berkeley, apresenta um suporte à modelagem, projeto e simulação de quaisquer sistemas baseados em atores, focando na concorrência, na execução em tempo-real e em sistemas embarcados. Ele é um framework de software escrito na linguagem Java com uma interface gráfica chamada Vergil. Seu principal diferencial em relação a outras ferramentas é o suporte a modelos heterogêneos. Ou seja, ele dá o suporte a vários modelos de computação interagindo num mesmo ambiente [5].

Outras ferramentas de simulação vêm sendo utilizadas [7] nos últimos anos para auxílio do ensino, porém o Ptolemy adaptado para a educação permitirá que alunos criem seus próprios ambientes de simulação em qualquer área do ensino onde atores possam ser aplicados. Isso fornece uma enorme flexibilidade à plataforma educacional em desenvolvimento em nossos laboratórios, chamada EducAtor, e a transforma numa ferramenta única onde alunos podem resolver exercícios de praticamente quaisquer disciplinas de seu interesse.

2. Utilização do Ptolemy como uma ferramenta educacional

Este projeto se baseia no uso do Ptolemy como uma ferramenta educacional, mas para isso há algumas barreiras a serem ultrapassadas: *i)* Ptolemy é um software para fins científicos e industriais. A configuração de ambientes utilizando atores previamente desenvolvidos não é muito difícil, mas

criar novos atores não é uma tarefa trivial, principalmente porque requer conhecimentos em desenvolvimento de sistemas utilizando Java; *ii)* Considerando que os atores tenham sido desenvolvidos por especialistas, o Ptolemy não oferece nenhum tipo de ajuda sobre como o aluno deverá montar seus ambientes; *iii)* Caso um professor configure um ambiente para que seja trabalhado por seus alunos, ele não terá controle sobre o que os alunos fizeram com o ambiente ou com os atores. Se um aluno, por exemplo, modificar a situação de um ator, o professor não saberá disso, muito menos poderá reverter o processo.

Essas limitações tornam o Ptolemy um sistema não preparado para o e-Learning, principalmente à distância. Porém, como ele é um sistema de código aberto, podemos fazer as modificações necessárias para torná-la uma ferramenta ainda mais poderosa. As soluções para essas limitações são apresentadas em detalhes nas próximas seções, mas podemos resumir para cada limitação citada, respectivamente, as seguintes estratégias são propostas: *i)* Para o ensino de matérias voltadas à computação, a estratégia é ensiná-los a identificar e desenvolver seus próprios atores didáticos, que serão utilizados por eles e pelos outros alunos que cursarão a disciplina nos semestres seguintes, formando um repositório de atores. Para o ensino de disciplinas fora da área da computação, os alunos não participarão da concepção dos atores diretamente, mas poderão apoiar os alunos de computação na concepção dos mesmos; *ii)* Adaptar o Ptolemy tornando sua interface mais amigável, com instruções em português, e para que instruções passo-a-passo sejam dadas aos alunos na montagem de seus ambientes na forma de atividades. Tais atividades serão escritas pelos professores auxiliados pelos alunos de licenciatura e o simulador será capaz de passá-las nos momentos oportunos aos alunos; *iii)* Adaptar o Ptolemy para que ele também possa verificar e passar para o professor, como o aluno está realizando (ou realizou) as atividades escritas pelo professor. Sendo assim, a presença do professor junto ao aluno não é mais essencial, já que a ferramenta será capaz de gerar um relatório sobre como cada aluno desenvolveu suas atividades.

3. Metodologia

Para executar tais estratégias com êxito, os primeiros passos já foram dados e são descritos neste trabalho através de uma metodologia para criação de atores e materiais didáticos.

Esta metodologia pode ser vista como a execução das etapas descritas com seus respectivos detalhamentos.

3.1. Decidir que nível de detalhamento será aplicado a cada ator

O nível de detalhamento diz respeito à quão semelhante o ator será do objeto real modelado. Numa memória, por exemplo, pode-se escolher modelá-la em nível de bits, de bytes, palavras ou transações. No nível de bit, o detalhamento é muito maior, então, cada bit dos barramentos deve ser modelado e sua respectiva ação de resposta implementada. Já no nível de transação, é apenas necessário saber que um dado será enviado para a memória através do barramento de dados, um endereço através do barramento de endereço etc.

3.2. Identificar e estudar possíveis atores

Faz parte do aprendizado do aluno, buscar compreender como o ambiente que ele deseja modelar funciona, que componentes devem se tornar atores e como esses atores funcionam em detalhes. Em nosso exemplo, onde a organização de uma CPU é utilizada, se fez necessário o entendimento de conceitos como ciclo de instrução, temporização e largura de barramentos. O nível de detalhamento escolhido na etapa anterior é decisivo na identificação dos atores. Quanto maior o nível de detalhamento, maior a quantidade necessária de atores. Num nível de bits serão necessários atores para modelar registradores e a lógica para ler bit a bit dos dados de entrada. Já num modelo de transações, a leitura de números completos pode ser feita de uma única vez (até na forma de Strings, para facilitar ainda mais sua interpretação).

3.3. Desenvolver os atores

Depois de tomadas as decisões sobre como os atores serão desenvolvidos, parte-se para o desenvolvimento propriamente dito. Basicamente, para cada porta de entrada deve ser estabelecida a reação respectiva para cada tipo de mensagem recebida, estabelecendo qual será a mensagem enviada através de cada porta de saída.

3.4. Simular, testar e corrigir os atores implementados

Após os atores terem sido desenvolvidos, eles devem ser combinados num cenário de simulação para que seus comportamentos sejam simulados e testados, principalmente no que diz respeito ao inter-relacionamento entre os atores. No momento de desenvolvimento de um ator, nada poderá ser assegurado sobre o comportamento dos outros atores no ambiente. Por isso, faz-se necessário instanciar cada ator e alocá-lo em cenários diversos (vide Figuras 1 e

2), simulá-lo e testá-lo de acordo com os comportamentos esperados.

3.5. Criar um roteiro didático para ser aplicado com os alunos

O fato de um ou mais cenários existirem não determinará que isso poderá ser aproveitado como uma ferramenta didática para os alunos. Em nossa metodologia, um roteiro didático deve ser criado, indicando o que o aluno deverá realizar para que o exercício seja considerado satisfatório.

Nessa primeira versão, o roteiro didático é formado utilizando os próprios recursos que a interface gráfica Vergil oferece aos usuários.

3.6. Aplicar roteiro didático com os alunos

A Figura 1 apresenta o roteiro didático desenvolvido como estudo de caso para mostrar como o Ptolemy pode ser utilizado com propósitos pedagógicos.

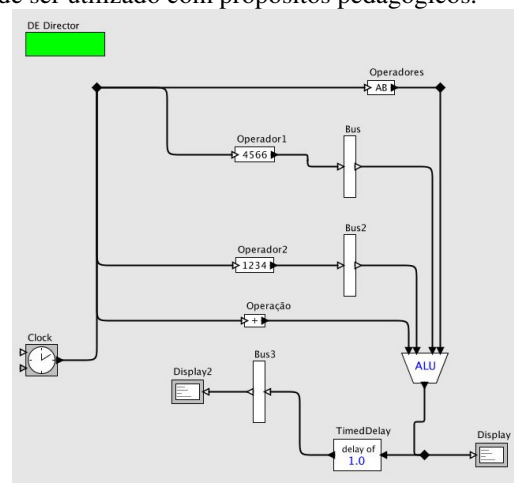


Figura 1. Exemplo de um exercício didático de funcionamento de CPU utilizando o Ptolemy.

Neste exemplo, o modelo de uma CPU está pronto e o aluno deverá configurá-lo para melhor funcionamento. Os parâmetros (os operadores e a operação) são passados para a ULA que executa a operação e apresenta o resultado na tela (*Display*). Como exercício o aluno deve ampliar este modelo, adicionando uma memória e automatizando este processo.

4. Experimento e resultados

Como um estudo de caso da aplicação, foi desenvolvido um exemplo no contexto do ensino de Organização de Computadores para alunos de Bacharelado em Sistemas de Informação e Licenciatura em Ciência da Computação, mais especificamente,

demonstrando conceitos básicos do funcionamento de uma Unidade Central de Processamento (CPU).

A Figura 2 apresenta como a CPU deverá ficar após sua configuração pelo aluno (esta é só uma das várias possíveis combinações). Para tal, quatro atores foram desenvolvidos especificamente para este roteiro didático. Os registradores, a ULA, o barramento (Bus) e o seletor. Os seletores servem para determinar que registradores servirão de entrada para a ULA. A partir disso, uma sintaxe para a instrução deve ser determinada pelo aluno, que será passada para os seletores. Cada seletor identificará parte da instrução e indicará que registrador ativar, que operação da ULA executar e em que registrador o resultado será salvo. Assim, temos a CPU completa que interpreta qualquer instrução definida pela sintaxe especificada e a executa.

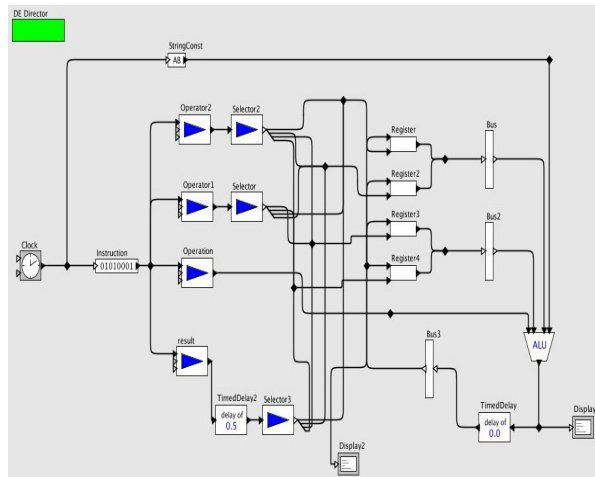


Figura 2. Exemplo completo da CPU.

Este exercício está sendo finalizado para também incorporar um ator que modela uma memória que já está implementada e em fase de teste. A partir deste momento será possível modelar e simular um sistema completo, onde um conjunto de instruções representando um programa poderá ser armazenado na memória e executado pela CPU projetada pelo aluno. Outro ponto importante deste exercício é o aprendizado do funcionamento e da importância da temporização de um sistema. Um sinal de *clock* deve ser criado (ator mais à esquerda na Figura 2), e atores especiais (disponibilizados pelo Ptolemy) devem ser utilizados para gerar um atraso inerente ao processo entre enviar uma instrução para a ULA e esperar seu resultado ficar pronto na saída.

4. Considerações finais

O simulador ainda está em desenvolvimento e espera-se um interesse por parte dos alunos pela possibilidade deles criarem seus próprios modelos (de CPU neste caso), e combinarem novos elementos e ampliarem seus estudos, tornando agora, elementos ativos na formação do conhecimento, onde eles próprios deverão buscar novas formas de criar os seus próprios modelos.

Atualmente utilizamos como bibliografia básica o livro *Arquitetura de Computadores* [8] por apresentar uma arquitetura hipotética própria para o ensino, estimulando os alunos a criarem suas próprias.

Nas próximas etapas o simulador fará a verificação automática da completude do roteiro didático e também apresentará uma interface mais amigável do simulador aos alunos.

Referências

- [1] Yurcik, W. e Gehring, E. F. A survey of web resources for teaching computer architecture. Proceedings of the 2002 Workshop on Computer Architecture Education (WCAE) - 29th International Symposium on Computer Architecture. ACM, Nova York, 2002.
- [2] Wikipedia. Actor Model, http://en.wikipedia.org/wiki/Actor_model.
- [3] Hewitt, C., Bishop, P., Stieger, R.: A Universal Modular Actor Formalism for Artificial Intelligence. Proceedings of the 1973 International Joint Conference on Artificial Intelligence, pp. 235-246 (1973)
- [4] Lee, E. A., "Computing Needs Time", Communications of the ACM, Vol. 52, Issue 5, May 2009.
- [5] Lee, E. A., Neuendorffer, S., "Tutorial: Building Ptolemy II Models Graphically", EECS Department, University of California, Berkeley, Technical Report No. UCB/EECS-2007-129, October 31, 2007.
- [6] Leung, M., Lee, E. A., "An Extensible Software Synthesis Framework for Heterogeneous Actor Models," in Proceedings of the 7th Model-driven High-level Programming of Embedded Systems Workshop (SLA++P 08), Budapest, Hungary, March 2008.
- [7] Aldrich, C.: Learning by Doing : A Comprehensive Guide to Simulations, Computer Games, and Pedagogy in e-Learning and Other Educational Experiences. San Francisco: Pfeifer — John Wiley & Sons (2003).
- [8] Delgado, J. e Ribeiro D., *Arquitetura de Computadores*, 2ª edição, Rio de Janeiro, LTC, 2009. ISBN:978-85-216-1660-3.