Executor de modelos baseados em fluxos e estoques

Daniel A. Moreira[1], Eric Eduardo da S. Alencar[2], Alexandre G. de Lima[3]

[1]Graduando de Tecnologia em Análise e Desenvolvimento de Sistemas – IFRN. e-mail: daniel_alves_09@hotmail.com
[2]Graduando de Tecnologia em Análise e Desenvolvimento de Sistemas – IFRN e bolsista PIBIC/IFRN. e-mail: eric_ed1@yahoo.com.br
[3]Diretoria Acadêmica de Gestão e Tecnologia da Informação – IFRN. e-mail: alexandre.lima@ifrn.edu.br

Resumo: O presente artigo apresenta um executor de modelos baseado em fluxos e estoques a ser incorporado no *software* de simulação de gestão empresarial SIMADM, que tem o objetivo de servir como ferramenta de apoio ao ensino da administração. Desenvolvida no fim dos anos 50 pelo engenheiro eletricista Jay Forrester, a Dinâmica de Sistemas é uma alternativa bastante eficaz na simulação de modelos dinâmicos com características de retroalimentação e de atrasos no fluxo de recursos. A falta de implementações livres, em linguagem Java, de executores de modelos baseados em fluxos e estoques estimulou o desenvolvimento realizado na pesquisa. Além de detalhes internos do executor desenvolvido, o texto também relata a simulação de três modelos utilizados durante os testes. Estes resultados levam ao entendimento de que o executor funciona adequadamente, apesar de que testes com modelos mais complexos ainda são necessários.

Palavras–chave: simulação, dinâmica de sistemas, fluxos e estoques.

1. INTRODUÇÃO

O SIMADM foi um *software* desenvolvido no ano de 2005 em um projeto financiado pela FINEP, executado pela UFRN, e que envolveu a participação de pesquisadores da UFRN, CEFET-RN (atual IFRN), UFRJ e UFPB. O objetivo principal do SIMADM é servir como uma ferramenta de apoio ao ensino da administração ao disponibilizar um ambiente de simulação de gestão empresarial.

O resultado final do projeto SIMADM foi considerado satisfatório, visto que além do desenvolvimento de um *software* com as características inicialmente planejadas, o SIMADM também foi utilizado como ferramenta de apoio didático no departamento de Administração da UFRN como descrito por Oliveira (2006) e Nascimento (2006).

Entretanto, os conceitos de simulação adotados para o SIMADM carecem de uma base teórica estabelecida. Tal característica ocasiona eventualmente em simulações com comportamento inesperado, dificultando a definição de testes e validações. Além disso, estes conceitos são desconhecidos da comunidade de pesquisadores de simulação empresarial, dificultando a adoção e compreensão do *software*. Assim, a equipe de desenvolvimento do SIMADM chegou à conclusão de que esta é a principal deficiência do SIMADM.

Diante disto, surgiu a necessidade de que o SIMADM adotasse um novo paradigma de simulação. Após um seminário de trabalho realizado entre os pesquisadores do projeto, o modelo de fluxos e estoques da dinâmica de sistemas destacou-se como um método de simulação adequado. Esta conclusão foi reforçada após a participação de integrantes do projeto no 1º Simpósio Internacional de Dinâmica de Negócios ocorrido na cidade de Brasília no ano de 2006. No entanto, questões pessoais dos participantes do projeto e a falta de financiamento fizeram com que os trabalhos fossem interrompidos até a sua retomada em 2012 como um projeto de iniciação científica.

A partir de então, o requisito mais importante para o SIMADM passou a ser a incorporação de modelos de simulação baseados em fluxos e estoques. Inicialmente foi realizada uma pesquisa sobre implementações livres, em linguagem Java, de executores de modelos existentes que pudessem ser incorporados facilmente ao SIMADM. Como a pesquisa não encontrou implementações com estas características, optou-se pela implementação de um *software* que permitisse a construção e execução de modelos baseados em fluxos e estoques.

ISBN 978-85-62830-10-5

Assim, este trabalho apresenta os resultados obtidos no desenvolvimento do executor de modelos que foi desenvolvido como componente de uma versão aprimorada do *software* SIMADM.

2. DINÂMICA DE SISTEMAS

A Dinâmica de Sistemas (*System Dynamics*) foi proposta e desenvolvida no fim dos anos 50 na Escola de Administração do Massachusetts Institute of Thecnology pelo Engenheiro Eletricista Jay Forrester. Chamada inicialmente de Dinâmica Industrial (FORRESTER, 1961), foi aplicada inicialmente no estudo da atividade industrial para mostrar como a construção organizacional e o fluxo de informações e de recursos impactam o desempenho de um empreendimento.

Com a expansão da teoria, Forrester aplicou os conceitos da Dinâmica Industrial em estudos estratégicos urbanos e mundiais, como descrito nos livros *Urban Dynamics* (Dinâmica Urbana) e *World Dynamics* (Dinâmica Mundial). A partir da constatação de aplicação diversa, a metodologia foi renomeada para Dinâmica de Sistemas. Atualmente são conhecidas aplicações da Dinâmica de Sistemas em estudos de áreas diversas como Física, Economia, Educação, Ecologia e Biologia.

Segundo Oliveira (2006), a Dinâmica de Sistemas busca entender e descrever como um sistema evolui ao longo do tempo. Este entendimento é auxiliado pela possibilidade de se testar diversos comportamentos dos elementos presentes no sistema. Assim, melhorias potenciais podem ser identificadas, dando suporte à tomada de decisões.

De acordo com Corbett Neto (2003), o foco da Dinâmica de Sistemas não está em valores numéricos precisos, mas sim nas tendências dinâmicas de um sistema. O que interessa é o seu comportamento ao longo do tempo, ou seja, se o sistema tende à oscilação, ao equilíbrio, ao crescimento ou ao declínio.

Ainda segundo Corbett Neto (2003), o pressuposto central da Dinâmica de Sistemas é de que o comportamento dinâmico do sistema é gerado pela sua estrutura causal, isto é, suas relações de causa e efeito. Uma das principais consequências destas relações é a ocorrência de laços de realimentação, chamados mais comumente de *feedbacks*. Um *feedback* pode ter impacto de reforço no sistema, ampliando o efeito de uma decisão (seja ela negativa ou positiva), ou de impacto regulador, levando o sistema ao equilíbrio.

Diagramas de enlace causal são utilizados para visualização qualitativa das relações de causa e efeito entre os elementos do sistema. Tais diagramas são úteis para a construção e discussão de uma situação-problema, permitindo uma comunicação fácil e direta (VILLELA, 2005). Um exemplo destes diagramas está ilustrado na Figura 1. Os símbolos positivos nas setas indicam um comportamento na mesma direção, ou seja, se a informação na origem da seta tem seu valor aumentado, a informação na ponta da seta também aumenta de valor. Se a origem diminui de valor, o que está na ponta também diminui. No caso de um símbolo negativo, o comportamento é o inverso.

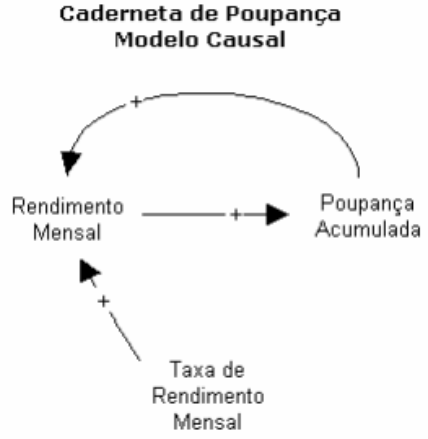

Figura 1 - exemplo de diagrama de enlace causal. Adaptado de Villela (2005).

ISBN 978-85-62830-10-5

Apesar de úteis na representação de relações de causa e efeito, diagramas de enlace causal não são úteis para descrever os relacionamentos em termos quantitativos. Para este fim, a Dinâmica de Sistemas trabalha com uma ferramenta conhecida como diagramas (ou modelos) de fluxos e estoques.

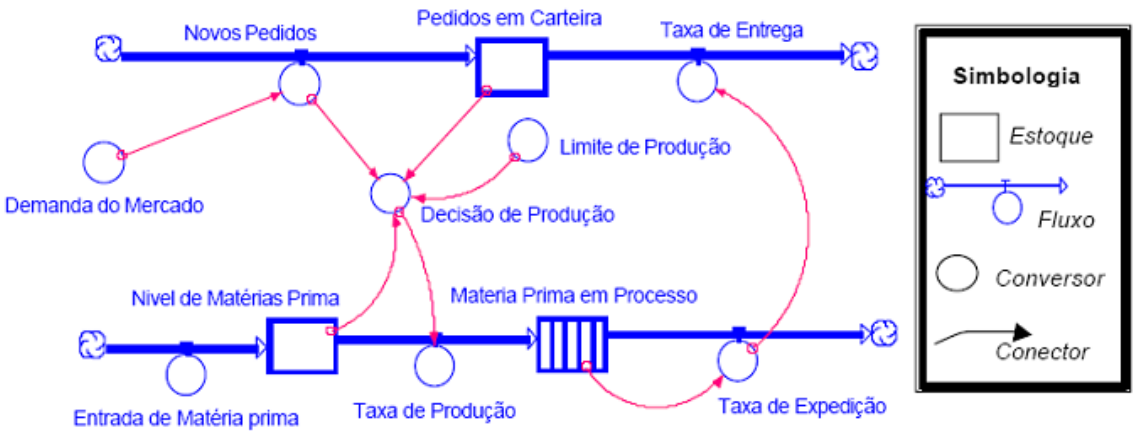

Figura 2 - exemplo de diagrama de fluxos e estoques gerado no *software* IThink.

Os elementos constituintes de um modelo de fluxos e estoques podem ser visualizados no diagrama da Figura 2 e estão descritos a seguir:

- **Estoques**: representam acumulações de recursos ao longo do tempo. Os recursos podem ser grandezas tangíveis (pessoas, materiais, água, etc.) ou intangíveis e abstratas (informação, pacotes de dados, conforto, popularidade, etc.).
- **Fluxos**: representam o transporte de recursos e sempre atuam sobre os estoques. Um fluxo pode atuar como alimentador ou consumidor de um estoque, ou simultaneamente como consumidor e alimentador de estoques distintos. Os fluxos são medidos em termos de vazão, ou seja, unidade do recurso por unidade de tempo (e.g., peças/hora, vendas/dia). A vazão de um fluxo pode ser um valor constante ou então ser determinada por uma equação. Em muitos modelos não há relevância na origem ou destino de um fluxo. Nestes casos, a maioria dos *softwares* representam o destino ou origem com uma nuvem.
- **Conversores ou variáveis**: representam constantes, parâmetros, equações ou decisões tomadas pelo usuário do modelo. Diferentemente dos estoques os conversores não acumulam.
- **Conectores**: estabelecem relações de influência entre os elementos de um sistema, indicando que outros elementos de modelo compõem a equação de cálculo de um elemento específico.

Após o estabelecimento de todos os elementos integrantes do modelo, é possível a realização de simulações a partir da sequência de passos presente no Quadro 1.

Fase de Inicialização 1. Crie uma lista de equações na ordem de avaliação necessária. 2. Calcule o valor inicial de todos os estoques, fluxos e conversores (em ordem de avaliação). Fase de Iteração 1. Estime a mudança nos estoques ocorrida durante o intervalo de tempo **dt**. 2. Baseado nessas estimativas calcule novos valores para os estoques.

ISBN 978-85-62830-10-5

3. Utilize os novos valores dos estoques para determinar novos valores para os fluxos e os conversores.
4. Adicione **dt** ao tempo de simulação.
5. Interrompa a iteração quando tempo >= tempo de parada.

Quadro 1 - algoritmo de simulação de um modelo de fluxos e estoques. Adaptado de Radzicki e Taylor (1997).

A Equação I mostra como estimar o valor de um estoque **E** no tempo **t** (passo 1 da fase iteração):

$$E(t) = E(t_0) + \int_0^t (\text{vazão de entrada}(t) - \text{vazão de saída}(t))dt \qquad \textbf{(I)}$$

Desta forma, um estoque pode ser descrito como uma soma discreta ou contínua dos seus fluxos de entrada e de saída (YAMAGUCHI, 1999). Infelizmente a solução analítica da equação I geralmente é difícil de obter. Uma alternativa para encontrar a solução da equação é determinar uma aproximação numérica da integral.

Um método de integração numérica bastante utilizado em Dinâmica de Sistemas é o método de Euler disponível na equação II, onde **dt** é uma fração ($0 <$ **dt** ≤ 1) de uma unidade de tempo da simulação. Quanto menor o valor de **dt**, mais precisa será a aproximação.

$$E(t) = E(t - dt) + (\text{vazão de entrada}(t - dt) - \text{vazão de saída}(t - dt))dt \qquad \textbf{(II)}$$

Outro método disponível comumente em *softwares* de simulação baseados em fluxos e estoques é a família de métodos de Runge-Kutta. Um método Runge-Kutta de segunda ordem realiza dois cálculos para cada fluxo no estoque, enquanto que um método de quarta-ordem realiza quatro cálculos. As equações III a V mostram como estimar o valor de um estoque para uma variação de tempo **dt** no método Runge-Kutta de segunda ordem.

$$\mathrm{F1} = dt * \text{fluxos}(t - dt) \qquad \textbf{(III)}$$
$$\mathrm{F2} = dt * \text{fluxos}(E(t) + \mathrm{F1}) \qquad \textbf{(IV)}$$
$$E(t) = E(t - dt) + 0.5 * (\mathrm{F1} + \mathrm{F2}) \qquad \textbf{(V)}$$

Basicamente, o método Runge-Kutta de segunda ordem estima a variação de um estoque com base na média dos fluxos que ocorrem ao início e término de cada ciclo de simulação (RADZICKI; TAYLOR, 1997).

3. EXECUTOR DE MODELOS BASEADO EM FLUXOS E ESTOQUES

A incorporação de modelos baseados em fluxos e estoques tornou-se o principal requisito do SIMADM 2 – a segunda geração do *software* desenvolvido em 2005. Nesta seção é apresentada a implementação produzida pelos autores, a qual pode ser considerada uma continuação do trabalho desenvolvido por Oliveira (2009).

Baseado nas diretrizes de baixo acoplamento e alta coesão do projeto orientado a objetos, além da carência de implementações livres em linguagem Java de executores de modelos baseados em fluxos e estoques, optou-se por um desenvolvimento em forma de API (*application programming interface*) que pudesse ser facilmente incorporada por outros projetos de *software* além do SIMADM. Este formato permite a adoção do executor de modelos em qualquer tipo de aplicação, não importando a sua interface com o usuário (e.g., *web*, *desktop*, *mobile*, etc.).

ISBN 978-85-62830-10-5

As classes do executor de modelos foram organizadas na estrutura de pacotes exibida na Figura 3.

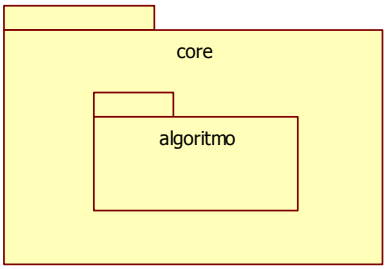

Figura 3 - estrutura de pacotes do executor de modelos.

No pacote core estão definidas classes de exceções, que representam erros relativos à lógica de simulação ou construção de modelos, e classes que representam os elementos constituintes de um modelo como fluxos e estoques. As principais classes do pacote estão ilustradas na Figura 4.

Ao interpretar o diagrama vê-se que uma instância de modelo será constituída por vários objetos do tipo componente de modelo. Entretanto, como as classes ComponenteDeModelo e ComponenteInfluenciavel são abstratas, um componente de modelo terá como tipo real a classe estoque, fluxo ou variável auxiliar.

Os conectores são representados pela associação, denominada influências, existente entre as classes ComponenteInfluenciavel e ComponenteDeModelo. Logo, a definição de um conector é realizada inserindo o componente de origem do conector (elemento influenciador) na coleção influências pertencente ao componente de destino do conector (elemento influenciado). Sendo assim, percebe-se que um estoque não pode ser definido como o destino de um conector. A justificativa desta opção de projeto é que um estoque deve ser alterado apenas pelos seus fluxos.

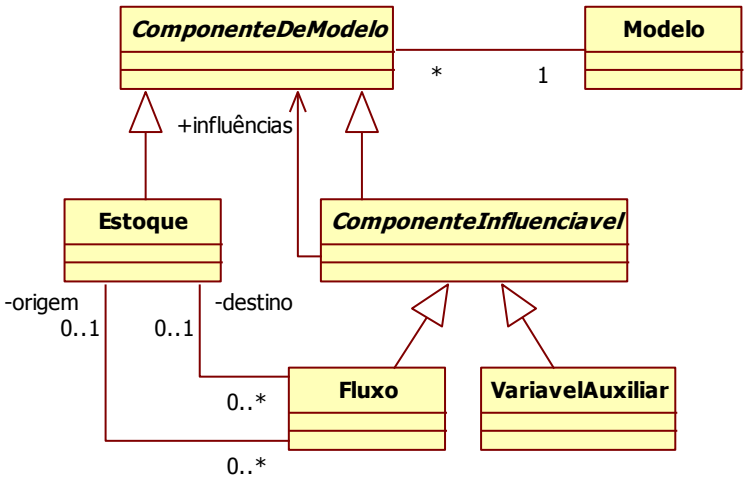

Figura 4 - principais classes do pacote core.

No pacote core.algoritmo encontram-se as classes responsáveis pela interpretação das equações dos componentes de modelo e que contém implementações dos métodos numéricos de integração utilizados no cálculo dos estoques. Estas classes podem ser visualizadas na Figura 5.

ISBN 978-85-62830-10-5

A interpretação das equações dos componentes de modelo é delegada para a API Jep 2.4 (SOURCEFORGE, 2012), uma biblioteca de código aberto desenvolvida, por terceiros, especificamente para esta finalidade. O diagrama ilustra este funcionamento através do relacionamento de dependência entre a classe Algoritmo e o componente jep 2.4.

A classe abstrata Algoritmo também possui a finalidade de definir as operações que devem ser implementadas pelas suas subclasses, representando, assim, um método genérico de integração numérica. Desta forma, cabe a cada subclasse definir o método de integração utilizado. Até o momento, estão implementados o método de Euler e o método Runge-Kutta de quarta ordem.

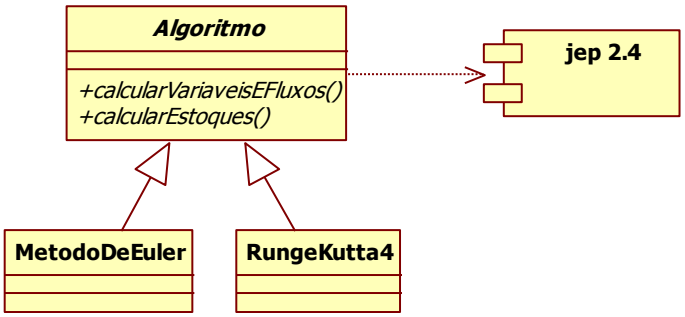

Figura 5 - classes do pacote core.algoritmo.

4. RESULTADOS

Para validar a implementação, foram realizadas simulações a partir dos seguintes modelos com resultados numéricos descritos na literatura: *population model* (STERMAN, 2000), modelo financeiro (VILLELA, 2005) e modelo de poupança (VILLELA, 2005). A Tabela 1 sumariza as características de cada modelo. Para todos os modelos, os valores obtidos com o executor implementado foram condizentes com os valores apresentados pelos respectivos autores.

Tabela 1 - características dos modelos utilizados na validação juntamente com os métodos de integração adotados.

Modelo	Estoques	Fluxos	Conversores	Conectores	Métodos de integração utilizados
Population	1	2	2	4	Runge-Kutta 4 Euler
Financeiro	2	4	12	18	Euler
Poupança	1	1	1	2	Euler

Em *population model*, a avaliação levou em conta a evolução do estoque e dos fluxos, visto que os conversores são constantes. Para este modelo as simulações apresentaram uma taxa média de erro igual a 0.00% para o método Runge-Kutta de 4ª ordem, como também para o método de Euler.

Para o modelo de poupança foi comparada apenas a evolução do estoque, já que a literatura de referência (VILELLA, 2005) não apresenta a evolução do fluxo. Nesta simulação também foi obtido o valor de 0.00% para a taxa média de erro.

Por ser relativamente mais complexo, o modelo financeiro surge como uma opção interessante para avaliação do executor de modelos. A complexidade adicional é justificada pela quantidade maior de componentes e de relacionamentos causais (conectores) e a utilização de valores aleatórios na geração do valor de três conversores. Consequentemente, esta característica impediu a comparação direta de valores numéricos, já que estes apresentam variação a cada simulação. Assim, o teste

ISBN 978-85-62830-10-5

consistiu em verificar se os valores dos conversores situaram-se entre um intervalo específico e se o comportamento geral do sistema, determinado pela evolução dos estoques, foi compatível com o descrito por Villela (2005). A validação dos valores de todos os conversores foi bem-sucedida, cujos valores sempre se mantiveram dentro dos intervalos esperados. A evolução dos dois estoques (denominados *Dívida* e *Estoque*) presentes no modelo pode ser visualizada nos gráficos presentes nas Figuras 6 e 7. Para gerar as curvas do executor de modelos, foi utilizada a média dos valores obtidos em cinco simulações. Na Figura 6 é possível observar de forma clara que a evolução da *Dívida* apresentou o mesmo comportamento. Na Figura 7, também é possível observar um comportamento semelhante na evolução do *Estoque*, apesar de que esta evolução não tenha sido tão uniforme no caso da *Dívida*.

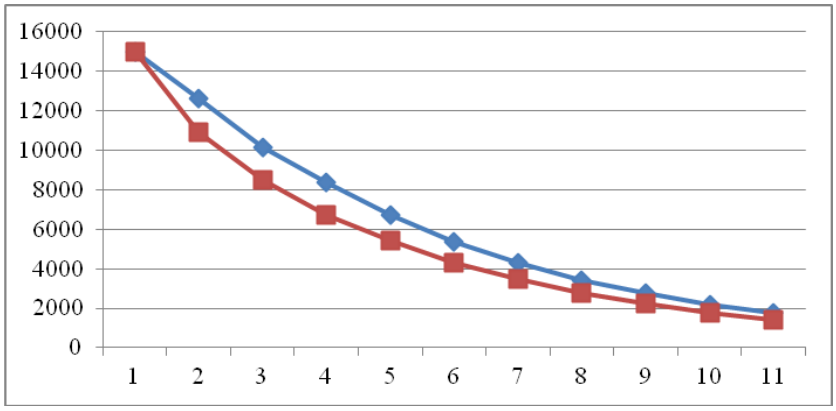

Figura 6 - evolução do estoque denominado *Dívida* no modelo financeiro. A curva azul representa o comportamento descrito por Villela (2005) e a linha vermelha o comportamento médio obtido em cinco simulações com o executor implementado.

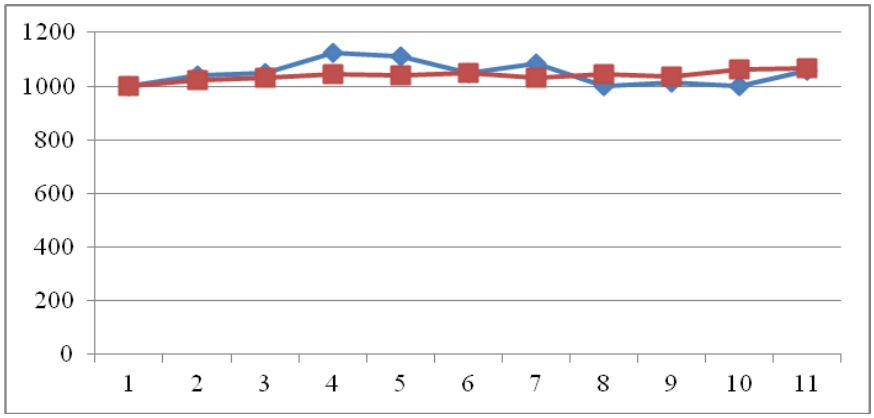

Figura 7 - evolução do estoque denominado *Estoque* no modelo financeiro. A curva azul representa o comportamento descrito por Villela (2005) e a linha vermelha o comportamento médio obtido em cinco simulações com o executor implementado.

5. CONCLUSÕES

Os resultados obtidos mostram que a implementação do executor de modelos funciona de forma precisa e satisfatória. Adicionalmente, mais modelos serão executados para que falhas sejam identificadas e consequentemente a confiabilidade na implementação seja melhorada. Para as futuras validações estão previstos modelos que incluam *delays* e modelos com elevada quantidade de componentes.

ISBN 978-85-62830-10-5

A adoção da biblioteca Jep foi uma decisão acertada, poupando um grande esforço que seria necessário na implementação de código que realizasse a interpretação de equações matemáticas. Outra consequência benéfica desta escolha foi a disponibilidade de funções matemáticas e lógicas disponibilizadas pela API e que, portanto, estão disponíveis no executor de modelos. Adicionalmente, a biblioteca Jep disponibiliza um mecanismo de extensão que torna fácil a adição de novas funções. Esta característica também é bem-vinda, pois está prevista a implementação de algumas funções normalmente disponíveis nos *softwares* de simulação baseados em fluxos e estoques (e.g., função *pulse*) e que ainda não são disponibilizadas pelo executor de modelos.

REFERÊNCIAS

CORBETT NETO, T. **Introdução à Dinâmica de Sistemas**. 2003. Disponível em <http://www.corbett.pro.br/introds.pdf>. Acesso em 12 mar. 2009.

FORRESTER, J. W. **Industrial Dynamics**. Cambridge: MIT Press, 1961. 479p.

NASCIMENTO, L. B. F.; ARAÚJO, M. V. P. Práticas de ensino e aprendizagem aplicadas na disciplina de administração de materiais do curso de Administração da UFPB. *In*: SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO, 13, 2006, Bauru. **Anais eletrônicos...** Bauru: UNESP, 2006. Disponível em: <http://www.simpep.feb.unesp.br/anais/anais_13/artigos/647.pdf>. Acesso em 03 ago. 2012.

OLIVEIRA, A. S. C. **Desenvolvimento de executor de modelos baseado em fluxos e estoques**. 2009. 39p. Monografia (Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, Natal, 2009.

OLIVEIRA, F. P. S. *et. al*. Aplicação da simulação empresarial no ensino da graduação. *In*: SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO, 13, 2006, Bauru. **Anais eletrônicos…** Bauru: UNESP, 2006. Disponível em: <http://www.simpep.feb.unesp.br/anais/anais_13/artigos/123.pdf>. Acesso em 03 ago. 2012.

RADZICKI, M. J.; TAYLOR, R. A. **Introduction to System Dynamics:** A systems Approach to Understanding Complex Policy Issues. U. S. Department of Energy's, 1997. Disponível em < http://www.systemdynamics.org/DL-IntroSysDyn/inside.htm>. Acesso em: 03 ago. 2012.

SOURCEFORGE. **Jep, version 2.4:** Java Math Expression Parser. Disponível em <http://sourceforge.net/projects/jep/>. Acesso em 28 ago. 2012.

STERMAN, J. D. **Business Dynamics:** Systems Thinking and Modeling for a Complex World. McGraw Hill, 2000. 1008p.

VILLELA, P. R. C. **Introdução à Dinâmica de Sistemas**. Universidade Federal de Juiz de Fora, 2005. Disponível em <https://sites.google.com/a/engenharia.ufjf.br/ds/arquivos>. Acesso: 03 ago. 2012.

YAMAGUCHI, K. Stock-Flow Fundamentals, Delta Time (DT) and Feedback Loop: From Dynamics to System Dynamics. *In*: INTERNATIONAL CONFERENCE OF THE SYSTEM DYNAMICS SOCIETY, 17, 1999, Wellington. **Proceedings...** Wellington: System Dynamics Society, 1999. Disponível em <http://www.dis.osaka-sandai.ac.jp/~kaoru/Papers/NewZealand.pdf>. Acesso em 28 ago. 2012.

ISBN 978-85-62830-10-5

ISBN 978-85-62830-10-5