

ANÁLISE DE UM SISTEMA A EVENTOS DISCRETOS UTILIZANDO UM SIMULADOR COM CÓDIGO LIVRE

Túlio Almeida Peixoto

Universidade Candido Mendes - UCAM-Campos
Rua Anita Pessanha, 100, Campos dos Goytacazes, RJ, Brasil
tulioap@gmail.com

João José de Assis Rangel

Universidade Candido Mendes - UCAM-Campos
joao@ucam-campos.br

Ítalo de Oliveira Matias

Universidade Candido Mendes - UCAM-Campos
italo@ucam-campos.br

Arthur Zopellaro Soares

Instituto Federal Fluminense - IFF-Campos-Centro
Rua Dr. Siqueira, 273, Campos dos Goytacazes, RJ, Brasil
arthurazsoares@gmail.com

RESUMO

O presente trabalho apresenta uma análise por simulação computacional de um típico sistema a eventos discretos, utilizando o ambiente de simulação Ururau. O ambiente Ururau utiliza como base a biblioteca de simulação *Java Simulation Library*. O uso deste simulador discreto em software de código aberto e livre de custos possibilitou a compreensão, em maiores detalhes, da estrutura de funcionamento deste tipo de software e a construção de componentes não previstos em simuladores comerciais. Os resultados mostraram a possibilidade de modelagem de aspectos específicos do sistema, com algoritmos construídos nas camadas inferiores do software, bem como a manipulação de módulos lógicos do ambiente.

PALAVRAS CHAVE. Simulação a eventos discretos, Java, Código livre.

Área principal: SIM – Simulação.

ABSTRACT

This paper presents an analysis by computer simulation of a typical discrete event system, using the simulation environment Ururau. The environment Ururau uses as a basis the simulation library *Java Simulation Library*. The use of this discrete simulator in open source software and free of cost allowed us understanding, in more detail, the structure of operation of this type of software and the construction of components not expected in commercial simulators. The results showed the possibility of modeling of specific aspects of the system, with algorithms built into the lower layers of software, as well as the manipulation of logic modules of the environment.

KEYWORDS. Discrete event simulation. Java. Open source.

Main area: SIM – Simulation.

1. Introdução

O presente trabalho tem por objetivo apresentar uma análise por simulação computacional de um típico sistema a eventos discretos, utilizando um ambiente de simulação com código livre. O ambiente de simulação é o Ururau, proposto inicialmente por Peixoto, Rangel e Matias (2011). Trata-se do primeiro ambiente para construção de modelos de simulação a eventos discretos elaborado no Brasil.

O ambiente de desenvolvimento do Ururau utiliza como base a biblioteca de simulação *Java Simulation Library* (JSL), proposto por Rosseti (2008). O JSL permite a construção de modelos orientados a processo e, quando necessário, a adição de novos comandos, caso o modelo de simulação exija maior grau de complexidade. Também utiliza a linguagem Java, que é multiplataforma. O software desenvolvido tem licença GPL (General Public License), o que permite desenvolvimento em código livre.

O uso de um ambiente de simulação a eventos discretos em software de código aberto e livre de custos possibilita a compreensão, em maiores detalhes, da estrutura de funcionamento de um simulador. Também é possível a construção de componentes não previstos em simuladores comerciais e, ainda, entender e manipular as ferramentas de checagem de modelos, dentre outros aspectos comuns existentes nos ambientes comerciais de simulação discreta.

De modo geral, a utilização do software Ururau permite o desenvolvimento de modelos de simulação via interface gráfica, com a manipulação de módulos lógicos, utilizando uma simbologia baseada em IDEF-SIM, proposta por Montevechi et al. (2010). Por outro lado, podem-se construir, também, modelos em código Java, utilizando, para isso, a camada mais inferior do software, em JSL, e assim elaborar algoritmos específicos para um determinado modelo, conforme a necessidade.

Este artigo está organizado de forma que o leitor possa acompanhar as etapas comuns de uma típica análise de um sistema a eventos discretos e poder utilizar como exemplo para construção de outros modelos com o ambiente Ururau. Desta forma, os itens seguem uma estrutura metodológica adaptada de Banks (2010), da seguinte maneira: O item 2, a seguir, descreve a formulação do problema, com os dados necessários à compreensão do funcionamento do sistema em análise; o item 3, modelagem, apresenta o modelo conceitual com a descrição das funções que representam a dinâmica e as regras operacionais do sistema, além da verificação e validação do modelo; o item 4 auxilia a instalação e uso do software Ururau; o item 5 apresenta o modelo de simulação desenvolvido em Ururau, com detalhes da interface gráfica e de algoritmos em JSL; o item 6 apresenta alguns experimentos básicos realizados com o modelo. Por fim, o item 7 apresenta as conclusões do trabalho.

2. Descrição do Sistema

O sistema em análise é um posto de combustíveis, utilizado para abastecimento de veículos. Este é um típico sistema de filas a eventos discretos, semelhante a diversos outros, como linhas de montagens, sistemas logísticos de transporte de cargas, postos de atendimentos, etc. O objetivo da análise desenvolvida por simulação computacional com o software Ururau é o de compreender a dinâmica do sistema e, assim, poder avaliar a alocação de recursos e o fluxo operacional. Desta forma, o que se pretende demonstrar neste trabalho é a análise da dinâmica das operações do posto de combustíveis através da avaliação da necessidade de contratação de mais de um frentista com o aumento de 35% do fluxo de veículos.

No posto, há quatro bombas: Duas de um bico e duas de dois bicos. Cada bico é um tipo de atendimento, por exemplo, o bico 1 é para pagamento à vista de gasolina, já o bico 3 é para pagamento a prazo de gasolina e, por isso, há um tempo maior de atendimento, já que o frentista deve emitir manualmente uma nota com a placa do veículo, o nome do cliente e a quantidade de produto para o cliente. Assim, o processo de atendimento pode levar mais tempo em alguns bicos, mas em outros não. Em cada turno, é utilizado somente um frentista. A distância entre as bombas é de cerca de 2 metros. Assim, pode-se resumir a operação do sistema da seguinte forma: O bico 1 e 2 é de gasolina para pagamento à vista; o bico 3 é de gasolina a prazo; o bico 4 é de álcool para pagamento à vista ou a prazo; o bico 5 é de diesel à vista; e o bico

6 é de diesel a prazo. A descrição física do sistema está representada na Figura 1.

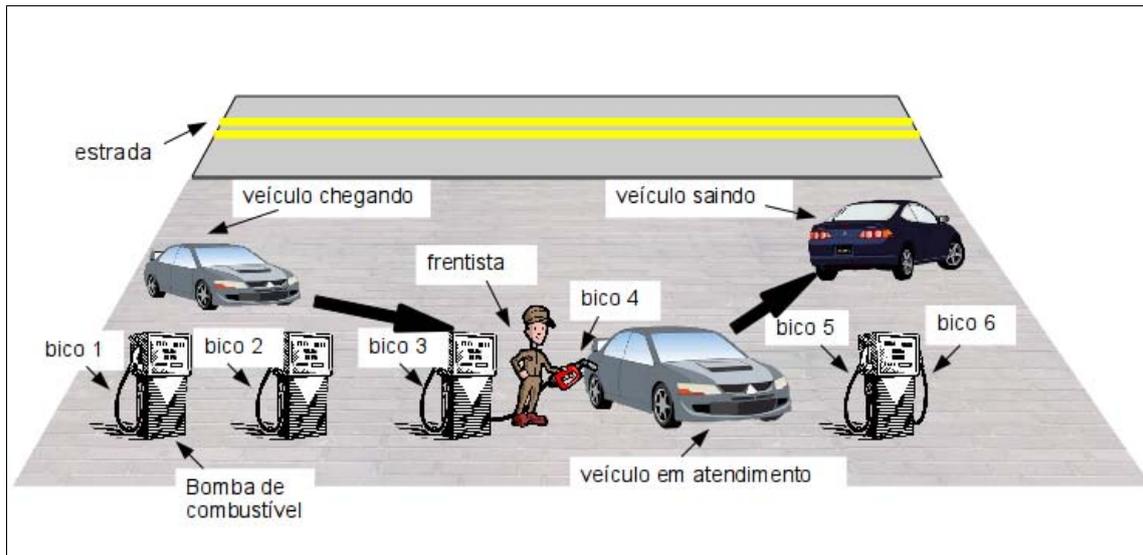


Figura 1 – Representação visual do posto de combustíveis modelado.
(Adaptado de PEIXOTO e RANGEL, 2010).

O levantamento dos dados foi realizado às sextas-feiras, sábados e domingos pelo turno da manhã (das 6h às 12h), entre 20 e 13 de dezembro de 2009. Os dados utilizados para a elaboração do presente estudo foram obtidos a partir de Peixoto e Rangel (2010), onde se pode encontrar a descrição mais detalhada deste sistema. Foram contabilizados, com a precisão de segundos, o tempo de chegada, o tempo de início do serviço, o tempo de fim do serviço e o número do bico da bomba para cada veículo atendido. Assim, foram calculados o tempo entre chegadas e o tempo do serviço. Uma amostra com o registro de todos os eventos pode ser vista na Tabela 1.

O período de atendimento foi calculado levando-se em conta a diferença entre o momento de saída e o de atendimento. Assim, o período de abastecimento e o período de pagamento somado ficam iguais ao período de atendimento. Esta simplificação foi feita para facilitar o levantamento dos dados, que foi aferido manualmente.

Tabela 1 – Amostra com o registro dos eventos ocorridos no sistema
(Adaptado de: PEIXOTO e RANGEL, 2010).

Cliente	Tempo entre chegadas (min.)	Tempo de chegada no relógio (h.:min.:seg.)	Tempo do serviço (min.)	Tempo de início do serviço no relógio (h.:min.:seg.)	Número do bico	Tempo final do serviço no relógio (h.:min.:seg.)
1	2.68	09:26:28	1.83	09:26:40	2	09:28:30
2	3.77	09:30:14	1.12	09:30:45	4	09:31:52
3	7.50	09:37:44	1.00	09:37:55	2	09:38:55
4	2.68	09:40:25	1.20	09:40:31	5	09:41:43
5	14.10	09:54:31	0.95	09:54:38	1	09:55:35
6	1.60	09:56:07	1.82	09:56:20	5	09:58:09
7	2.38	09:58:30	0.45	09:58:40	2	09:59:07
8	1.50	10:00:00	0.25	10:00:05	2	10:00:20
9	3.78	10:03:47	0.70	10:03:58	1	10:04:40

3. Modelagem do sistema

O modelo de simulação do sistema foi desenvolvido orientado pela metodologia proposta por Banks (2010), com os seguintes passos específicos para este trabalho: Formulação e análise do problema; planejamento do projeto; formulação do modelo conceitual; coleta de macro-informações e dados; tradução do modelo; verificação e validação; projeto experimental; experimentação; interpretação e análise estatística dos resultados; comparação e identificação das melhores soluções; e documentação e apresentação dos resultados. O modelo foi construído considerando um período de tempo pré-estabelecido para o início às 6h e término das operações às 12h, caracterizando assim, o modelo como de um sistema terminal.

A modelagem conceitual, segundo Sargent (2007), foi constituída visando a facilitar a visualização e interpretação do sistema. Sendo assim, foi feito um esquema que representa sua dinâmica, bem como as funções existentes.

Este esquema é o modelo conceitual e foi feito em IDEF-SIM, proposto por Montevechi et al (2010). O modelo está na Figura 2 e as entidades, processos e recursos estão nas Tabelas 2, 3 e 4, respectivamente.

Em função dos dados obtidos, foram determinadas as distribuições de probabilidade de maior aderência para o tempo entre chegadas dos veículos e para o tempo de atendimento para cada bico. Verificou-se que os veículos chegam um de cada vez e o tempo de atendimento para cada bico está mostrado nas Tabelas 2 e 3. Definiu-se que o tempo da primeira chegada é 0 minuto.

As funções apresentadas nas Tabelas 2 e 3 foram obtidas com a ferramenta de teste de aderência e análise estatística de dados, incluído no próprio Ururau. A Figura 3 mostra uma tela desta ferramenta com as informações que podem ser obtidas durante a modelagem dos dados de entrada.

Inicialmente, conforme mostra a Figura 2, os veículos (E1) chegam a cada Exponencial (4,14) minutos. Em seguida, cada carro é atendido em F_i , que se altera em função do combustível e forma de pagamento (à vista ou a prazo), que é definido pelo número do bico e é utilizado como recurso o frentista (R1). Para tanto, antecedendo à função atendimento F_i , é tomada uma condição (elemento X) submetida a uma probabilidade P_i . Por exemplo, um carro tem a chance de 26,9% de ser atendido usando-se o bico 1. Após o atendimento, o veículo sai do sistema.

Tabela 2 – Entidade do sistema presente na Figura 2.

Entidade	Descrição	Tempo entre chegadas	Número máximo de chegadas
E1	Veículo	Exponencial com média de 4,14 min.	Infinito

Tabela 3 – Funções dos processos presentes na Figura 2.

Função	Descrição	Tempo de serviço
F1	Atendimento - bico 1	Lognormal: média de 0,91min e desvio padrão de 0,68min.
F2	Atendimento - bico 2	Exponencial: média de 0,98min.
F3	Atendimento - bico 3	Normal: média de 3,3 min e desvio padrão de 2,06min.
F4	Atendimento - bico 4	Lognormal: média 0,91min e desvio padrão de 0,78min.
F5	Atendimento - bico 5	Exponencial: média de 2,11min.
F6	Atendimento - bico 6	Gamma: com forma de 0,933 e escala de 2,04min.

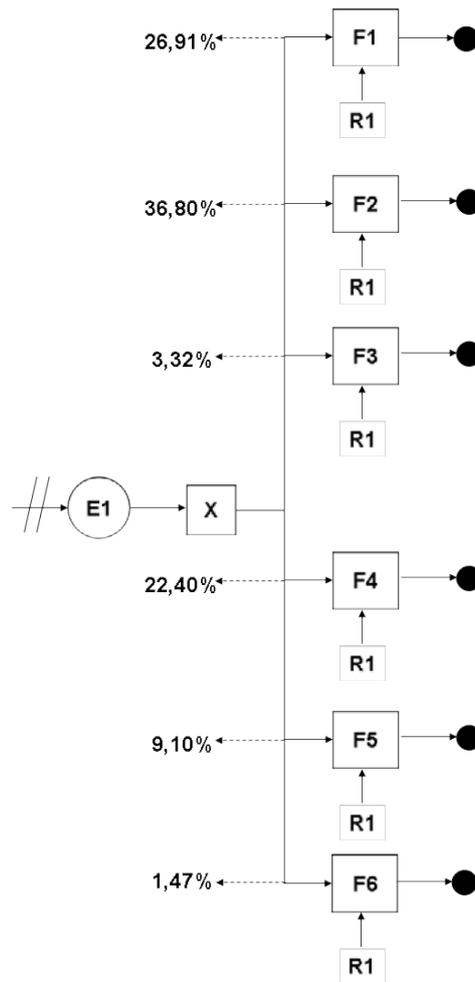


Figura 2 – Modelo conceitual do sistema em IDEF-SIM.

Tabela 4 – Recurso do sistema presente na Figura 2.

Recurso	Descrição	Quantidade
R1	Frentista 1	1

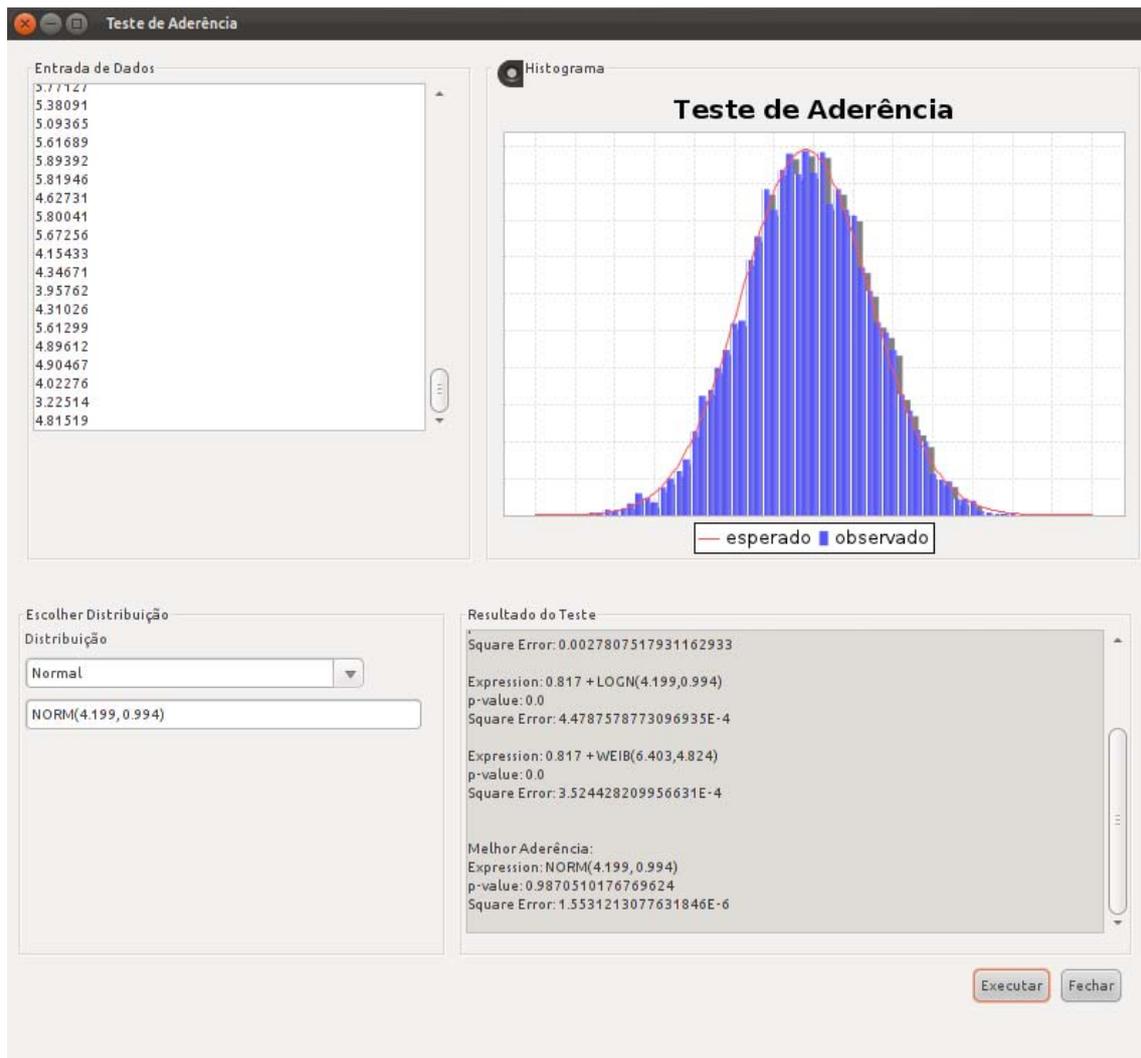


Figura 3 – Teste de aderência no ambiente Ururau.

4. Instalação do software Ururau

Para uso do ambiente de desenvolvimento do software Ururau pode-se acessar livremente em <https://bitbucket.org/tulioap/ururau>. Após baixar o arquivo compactado, descompacte-o em um diretório e execute o arquivo **ururau.jar**. Lembre-se que é necessário ter instalado na máquina o *Java Runtime Environment* (JRE) versão 6.0 ou superior, que pode ser obtido em <http://java.com/getjava>.

Para construir o modelo usando o código fonte, é preciso um ambiente de desenvolvimento - *Integrated Development Environment* (IDE), de preferência o NetBeans 7.0, que pode ser obtido em <http://www.netbeans.org>.

O ambiente do Ururau é composto internamente por um ou mais comandos de processo que estendem funcionalidades do JSL. No próprio endereço do software, pode-se baixar também o arquivo **README.TXT**, que apresenta uma explicação de como funciona cada comando.

5. Modelo de simulação

A Figura 4 apresenta o modelo de simulação em Ururau do sistema traduzido a partir do modelo conceitual mostrado na Figura 2. Note a semelhança do modelo de simulação e o modelo conceitual. Esta é uma facilidade que ocorre com o software Ururau, já que ele utiliza os módulos baseados na estrutura da linguagem IDEF-SIM.

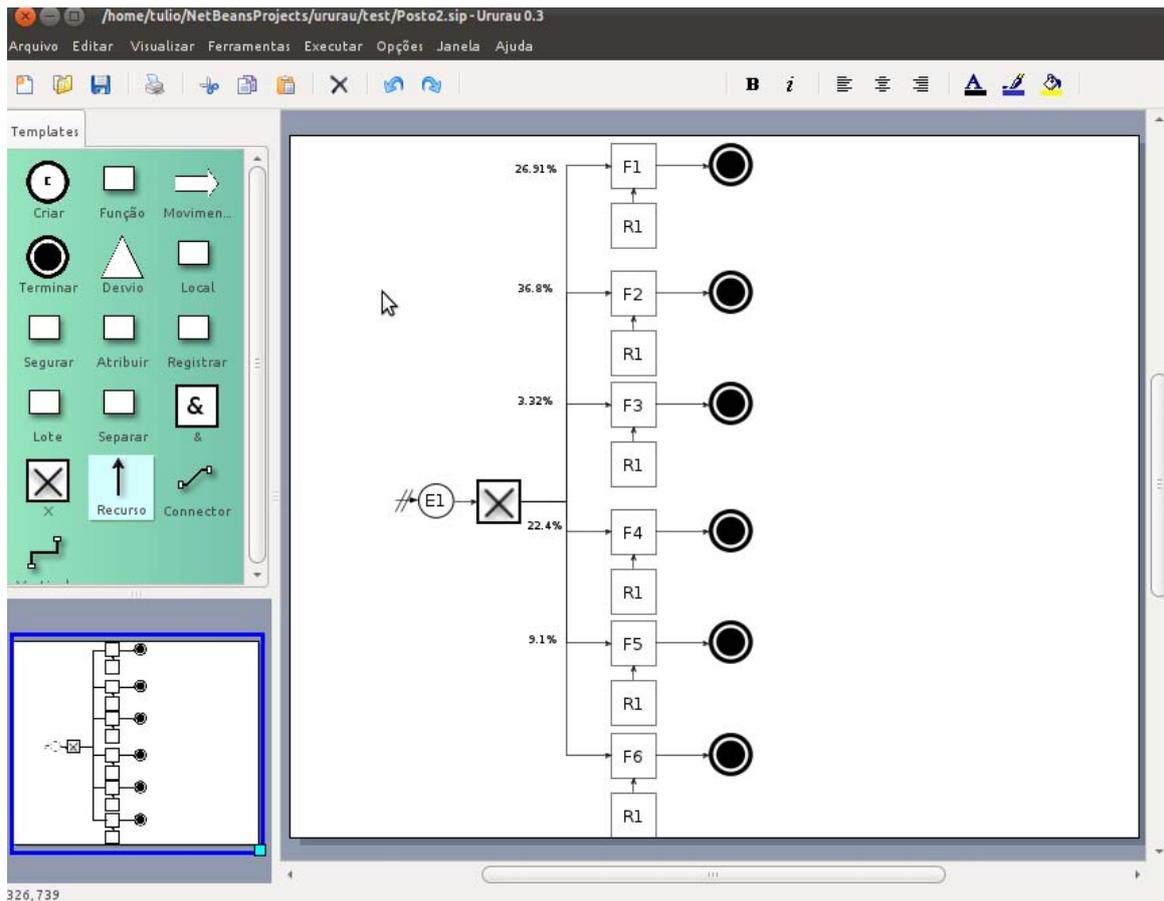


Figura 4 – Modelo de Simulação no ambiente Ururau.

Por outro lado, a construção de modelos de simulação no ambiente de desenvolvimento do software Ururau permite também a construção de algoritmos em JSL. A Figura 5 mostra uma parte do código do modelo em Java. Desta forma, o modelador pode trabalhar em ambas as camadas do software, de acordo com a necessidade exigida durante a modelagem do sistema.

```

41 public static Model buildModel(){
42
43 // criar modelo
44 Model m = Model.createModel();
45 RandomVariable tcServidor = new RandomVariable(m,
46                                     new Triangular(2,5,10), "processo Atendimento");
47 Resource r1 = new Resource(m, 3, "Atendente");
48 Queue q1 = new Queue(m, "Fila");
49
50 ProcessDescription mml = new ProcessDescription(m, "Modelo de Banco");
51 DistributionIfc tecCliente = new Exponential(2);
52
53 Variable v1 = new Variable(m, 1.0, "atendentes requisitados");
54 new EntityProcessGenerator(m, mml, tecCliente, tecCliente);
55 mml.addProcessCommand(new Process(m, v1, r1, q1,tcServidor));
56 mml.addProcessCommand(new Terminate(m,true,"Fim do servico"));
57
58 return m;
59
60 }

```

Figura 5 – Parte do modelo de simulação em código Java (JSL) no ambiente Ururau.

6. Experimentos e Análise dos Resultados

A estratégia de experimentação utilizada nas simulações computacionais foi do tipo Projeto Fatorial 2^k . Nesta estratégia, descrita em detalhes em Montgomery (2007), altera-se um dos dois níveis do fator (k) por vez, mantendo-se os demais fixos. A proposta foi iniciar os experimentos com uma configuração típica (a mesma utilizada na validação do modelo) para todos os fatores e depois se alterar os níveis de um fator de cada vez nos experimentos sucessivos.

Os Fatores (variáveis que podem ser controladas pelo modelo) foram o número de frentistas, o incremento da demanda e o incremento do atraso de atendimento. O incremento da demanda é o aumento do número de veículos em relação à situação de regime atual do posto, assim como o incremento do atraso de atendimento é o número que atrasa o processo do atendimento como, por exemplo, a contratação de um frentista inexperiente para realizar o atendimento.

A Tabela 5 descreve a relação dos fatores e níveis que foram atribuídos nos cenários simulados com o modelo. A utilização de dois níveis neste trabalho se justificou pela simplicidade obtida no projeto experimental e a possibilidade de se observar a continuidade unidirecional no comportamento das variáveis. Para o incremento da demanda, foram definidos dois valores. Para um aumento de demanda de aproximadamente 35% no verão (dados em função do histórico do faturamento do ano anterior), a taxa de entrada redefinida para Exponencial de 3,20 minutos de média. Já para o regime de funcionamento normal do posto é continua Exponencial de 4,14 min. de média. Isto porque o incremento é em relação ao frentista já contratado pelo posto. Caso se contrate um novo funcionário como frentista sem experiência, o atraso será de mais 25%, ou seja, o incremento será de 0,25 no processo de atendimento.

Tabela 5 – Relação de Fatores e Níveis atribuídos ao modelo.

Fatores	Nível 1	Nível 2
Número de frentistas	1	2
Demanda	Normal	Aumento
Atraso no atendimento	0	0,25

A Tabela 6 descreve os cenários que foram utilizados nas simulações computacionais com 3 fatores e 2 níveis em cada um deles, resultando em um total de 8 cenários.

Tabela 6 – Descrição dos cenários simulados.

Cenário	Número de frentistas	Demanda	Incremento do atraso do atendimento
1	1	Normal	0
2	1	Normal	0,25
3	1	Aumento	0
4	1	Aumento	0,25
5	2	Normal	0
6	2	Normal	0,25
7	2	Aumento	0
8	2	Aumento	0,25

A Tabela 7 apresenta os resultados obtidos com as simulações computacionais. A

variável de resposta definida para o experimento com o modelo foi o tempo de espera máximo (em minutos) para a fila do bico 2, pois é onde passa mais carros (maior porcentagem).

Tabela 7 – Resultados das simulações computacionais.

Cenário	Numero máximo de veículos no posto	Tempo de espera máximo (min.)
1	4	5,48
2	5	7,74
3	5	7,28
4	7	10,41
5	3	0,91
6	4	1,47
7	4	1,58
8	4	2,45

Note que, contratando mais um frentista antes da demanda aumentar (cenário 5 e 6 na Tabela 6), diminui-se o número máximo de veículos no posto de 4 (cenário 1) para 3 (cenário 5). O número máximo de veículos no posto é um indicativo na qualidade do atendimento já que, se o cliente vir mais do que 4 veículos e um frentista para atender todos, é bem capaz de ele nem parar no posto para abastecer o seu veículo.

Os cenários 2 e 4 (Tabelas 6 e 7) representam uma situação extrema onde existe a substituição do frentista já contratado por um inexperiente. Isto causa um significativo aumento no tempo médio de espera (entre 81% e 210%) comparado ao cenário 1. Também o número máximo de veículos sobe de 4 para até 7 (cenário 4 na Tabela 7) desestimulando, assim, um novo cliente a entrar no posto.

Note ainda que os cenários com um só frentista (cenários 1, 2, 3 e 4 na Tabela 6) resultam (cenários 1, 2, 3 e 4 na Tabela 7) em um tempo de espera máximo muito grande (entre 5,48 a 10,41 minutos). Quando se contrata mais um frentista (cenários 5, 6, 7 e 8 na Tabela 5), mesmo que inexperiente, o tempo de espera máximo para ser atendido cai significativamente para no máximo 2,45 minutos (cenário 8 na Tabela 7). Já o tempo de espera máximo para todos os cenários 5, 6, 7 e 8 cai para no máximo 2,45 minutos, ou seja, um tempo razoável de espera.

7. Conclusão

O trabalho apresentou uma descrição em forma de guia para o uso do ambiente de simulação a eventos discretos Ururau. Foi apresentada a modelagem e simulação de um típico sistema a eventos discretos. De acordo com a organização do trabalho, puderam-se acompanhar os passos que são normalmente utilizados para a construção de modelos de simulação discreta e utilizá-los como exemplo para elaboração de novos modelos.

Durante a descrição da modelagem do sistema, foram apresentados detalhes específicos, relativos à utilização do ambiente Ururau, como a manipulação dos módulos lógicos e aspectos da implementação de algoritmos em Java, nas camadas mais inferiores do software. Foi mostrada ainda a possibilidade de se poderem construir módulos específicos para um modelo em particular, de acordo com a necessidade do modelista. Estes são fatores potenciais que diferenciam a utilização de um software aberto e livre de custos.

Destaca-se, por último, que a criação do Ururau teve como objetivo principal o de contribuir para facilitar a difusão, o uso e a compreensão no Brasil da simulação discreta, desde a sua aplicação prática até a concepção interna de sua estrutura computacional e código fonte. Fica claro então que não se pretendeu aqui fazer comparações do software com os produtos comerciais

do ponto de vista de capacidade em se resolver problemas. A questão está centrada na formação de pessoas aptas a poderem utilizar a simulação discreta e os softwares de simulação de forma mais sólida e ampla.

Agradecimentos

Os autores gostariam de agradecer ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e à Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) pelo suporte financeiro para esta pesquisa.

Referências

- Banks, J.; Carson, J. S.; Nelson, B. L.; Nicol, D. M.** *Discrete-event system simulation*. 5nd ed. New Jersey: Prentice Hall, 2010.
- Montgomery, D.C.** *Design and analysis of experiments*, 7th Edition, John Wiley and Sons, 2009.
- Montevechi, J.A.B.; Leal, F.; de Pinho, A.F.; da Silva Costa, R.F.; de Oliveira, M.L.M.; da Silva, A.L.F.** (2010) Conceptual modeling in simulation projects by mean adapted IDEF: An application in a Brazilian tech company. *In: Winter Simulation Conference, 2010, Baltimore. Proceedings of the 2010 Winter Simulation Conference, 2010.* 1624-1635.
- Sargent, R. G.** (2007) Verification and validation of simulation models. *In: Winter Simulation Conference*, 124-137.
- Peixoto, T. A. ; Rangel, J. J. A.** (2010) Modelo de simulação para análise do funcionamento de um posto de combustíveis. *In: Encontro Mineiro de Engenharia de Produção, Coronel Fabriciano. VI EMEPRO.*
- Peixoto, T. A. ; Rangel, J. J. A. ; Matias, I. O.** (2011) Ururau - Um Ambiente de Simulação a Eventos Discretos. *In: XLIII SBPO. Ubatuba - SP.*
- Rosseti, M.D.** (2008) Java Simulation Library (JSL): an open-source object-oriented library for discrete-event simulation in Java. *In: Int. J. Simulation and Process Modelling*, Vol. 4, No. 1, 69-87.