

## MODELOS DE SIMULAÇÃO A EVENTOS DISCRETOS COM ASPECTOS DE DECISÃO HUMANA: UMA APLICAÇÃO COM O URURAU

**David Vasconcelos Corrêa da Silva**

Instituto Federal Fluminense (IFF)

[davidcefet@yahoo.com.br](mailto:davidcefet@yahoo.com.br)

**João Jose de Assis Rangel**

Universidade Candido Mendes (UCAM-Campos)

[joao@ucam-campos.br](mailto:joao@ucam-campos.br)

**Ítalo de Oliveira Matias**

Universidade Candido Mendes (UCAM-Campos)

[italo@ucam-campos.br](mailto:italo@ucam-campos.br)

**Dalessandro Soares Vianna**

Universidade Federal Fluminense (UFF)

[dalessandro@ucam-campos.br](mailto:dalessandro@ucam-campos.br)

**Túlio Almeida Peixoto**

Universidade Candido Mendes (UCAM-Campos)

[tulioap@gmail.com](mailto:tulioap@gmail.com)

### Resumo

Este trabalho tem como objetivo empregar redes neurais artificiais para representar de modo mais realístico ações provenientes de pessoas em modelos de simulação. Para isto, um módulo com uma rede neural artificial treinada (chamado de módulo inteligente) substituiu uma típica operação lógica de decisão tradicionalmente utilizada nos modelos de simulação discreta. A respectiva decisão foi avaliada em uma parte de um modelo de simulação hipotético onde ocorreria uma intervenção proveniente da ação de uma pessoa em uma situação normalmente típica em sistemas a eventos discretos. Para implementar o modelo de simulação com o módulo inteligente, foram usados o ambiente de desenvolvimento de modelos de simulação a eventos discretos Ururau e o software de criação e treinamento de redes neurais artificiais JAVANNS. Os resultados obtidos demonstraram a possibilidade do uso da abordagem proposta para representar certos aspectos de decisões humanas de forma mais realística. Tal afirmativa pôde ser avaliada com a similaridade dos resultados dos experimentos que compararam diferentes maneiras utilizadas para realizar decisões nos modelos de simulação discreta construídos.

**Palavras-chave:** Simulação a eventos discretos, redes neurais artificiais, Ururau, decisão.

### Abstract

This work aims to employ artificial neural networks in order to represent more realistic actions from people in simulation models. For this, a module with a trained artificial neural network (called intelligent module) replaced a typical logic operation of decision traditionally used in discrete simulation models. The decision was evaluated in a part of a hypothetical simulation model which would occur an intervention from the action of a person in a situation usually typical in systems discrete event. To implement the simulation model with the intelligent module, we used the development environment of discrete event simulation models Ururau and the software of creation and training of artificial neural networks JAVANNS. The results showed the possibility of using the proposed approach to represent certain aspects of human decisions more realistically. Such statement could be evaluated by the similarity of the results of experiments comparing different ways used to make decisions in discrete simulation models built.

**Key words:** Discrete event simulation, artificial neural networks, Ururau, decision.

### 1 INTRODUÇÃO

A simulação a eventos discretos é comumente empregada para analisar sistemas envolvendo aplicações em manufatura e logística. Isto ocorre, pois os ambientes de desenvolvimento de modelos de simulação permitem uma modelagem rápida e com alto grau de detalhes de aspectos relacionados à natureza dinâmica e estocástica de tais sistemas. Assim, muitas empresas vêm buscando implantar melhorias em seus processos produtivos, através da experimentação computacional em modelos de simulação cada vez mais realísticos (BANKS *et al*, 2010).

Na busca pelo desenvolvimento de modelos de simulação mais próximos à realidade dos sistemas representados, alguns pesquisadores vêm buscando formas de melhorar a modelagem com esta abordagem, ressalta Zülch (2006). Ou seja, gradativamente, aumenta-se o interesse pela modelagem de decisões humanas de modo mais detalhado e realístico. Contudo, grande parte dos modelos de simulação ainda representa apenas os aspectos mais lógicos e simples das ações provenientes de decisões humanas (ROBINSON *et al*, 2001).

Nos casos onde somente os aspectos lógicos das decisões são modelados, normalmente são utilizados os tradicionais operadores lógicos, como: “<”, “>”, “==”, “AND”, “OR”, “XOR” e suas derivações. Além disso, em outros casos, utilizam-se apenas probabilidades obtidas em dados históricos para representar as decisões humanas. Essas formas de representação são muitas vezes simplistas e pouco realistas (ZÜLCH, 2006).

Criar modelos de simulação com representações realísticas das decisões humanas é uma tarefa extremamente difícil segundo Zhao *et al* (2005). Para Robinson *et al* (1998), a dificuldade de programar modelos de simulação com representação de decisões humanas se deve basicamente a dois fatores: Complexidade do pensamento humano e necessidade de utilização de recursos não triviais nos softwares de simulação.

Segundo Robinson *et al* (2001), a maioria dos softwares de simulação não fornecem um conjunto necessário de funções que seriam úteis para modelar o comportamento humano. Além disso, na maioria dos softwares comerciais de simulação, o desenvolvimento de novos recursos é não trivial ou não permitido. Sendo assim, o uso de softwares de simulação que permitam a inserção de novos algoritmos e criação de novos recursos podem viabilizar a construção de modelos de simulação que considerem aspectos mais realísticos das tomadas de decisões provenientes da ação de uma pessoa.

Diante deste contexto, este trabalho propõe a utilização de redes neurais artificiais para representar o comportamento de ações de pessoas em modelos de simulação a eventos discretos, nos casos onde normalmente não é possível determinar os aspectos lógicos da tomada de decisão. Esta proposta foi implementada através da criação de um módulo com aspectos de inteligência humana, capaz de representar, de forma mais realística, as decisões realizadas por pessoas, acopladas a um ambiente de simulação a eventos discretos.

### 2 LINGUAGENS, AMBIENTES E BIBLIOTECAS USADAS EM SIMULAÇÕES A EVENTOS DISCRETOS

Modelos de simulação a eventos discretos podem ser construídos usando linguagens de programação de propósito geral (LPPG), como FORTRAN, C ou Java. Podem, ainda, ser construídos usando as linguagens específicas de simulação, como SIMAN ou GPSS (LAW, 2007; BANKS *et al* 2010). Por outro lado, os modelos podem também serem construídos usando os ambientes de desenvolvimento de modelos de simulação, como Arena, Promodel, Simul8 ou tantos outros (LAW, 2007; BANKS *et al* 2010; CHWIF & MEDINA, 2010).

Segundo Sargent (2011), a construção de modelos de simulação usando linguagens de simulação resulta em menos erros e na redução do tempo de construção do modelo, contudo, as LPPG oferecem maior flexibilidade ao modelador. Quanto aos modelos implementados em ambientes de simulação, Law (2007) afirma que possuem um custo menor, são mais fáceis de modificar e de detectar erros quando comparados com modelos construídos em LPPG ou

linguagens de simulação.

Pode-se dizer que os ambientes de simulação e as linguagens de simulação possuem diversas vantagens sobre as LPPG na construção de modelos de simulação. Contudo, há situações onde as LPPG são preferíveis. Em alguns projetos de simulação, elas são adotadas, principalmente, devido à maior flexibilidade e ao menor tempo de execução computacional do modelo. Atualmente, diversos modelos de simulação ainda são implementados com LPPG, principalmente em aplicações militares (LAW, 2007).

As LPPG podem ainda ser potencializadas por bibliotecas de simulação, como DSOL, JSL ou JAVASIM, que diminuem o tempo gasto na criação de modelos e reduzem o número de erros. As bibliotecas de simulação são um conjunto de rotinas que oferecem recursos específicos de simulação para as LPPG, tornando-as equivalentes às linguagens de simulação, mas sem a interface gráfica e as facilidades existentes nos ambientes de simulação (JACOBS *et al* 2002; ROSSETTI, 2008; UCHÔA JÚNIOR, 2006).

Criar um modelo de simulação capaz de associar as facilidades de uso dos ambientes de simulação com a flexibilidade das LPPG e também com as linguagens específicas de simulação e, ainda, possibilitar a inserção de novos algoritmos não é uma tarefa trivial. Assim, diante da necessidade de entender o funcionamento interno dos ambientes de simulação e de explorar novos algoritmos, pesquisadores têm sido motivados a desenvolverem um ambiente de simulação a eventos discretos com recursos gráficos, código aberto e alta flexibilidade, como o Ururau (PEIXOTO *et al*, 2011).

### 3 O AMBIENTE URURAU

O Ururau é um ambiente de desenvolvimento de modelos de simulação a eventos discretos construído com código aberto e distribuído sob licença de software livre, o que permite o uso acadêmico, a exploração de algoritmos internos e a criação de novas funcionalidades. Este ambiente utiliza recursos gráficos para a criação de modelos de simulação inspirados na simbologia da linguagem de modelagem IDEF-SIM, proposta por Montevechi *et al* (2010). A Figura 1 apresenta a interface gráfica do Ururau e seus recursos básicos: Menu principal; barra de ferramentas; elementos de modelagem; e navegador.

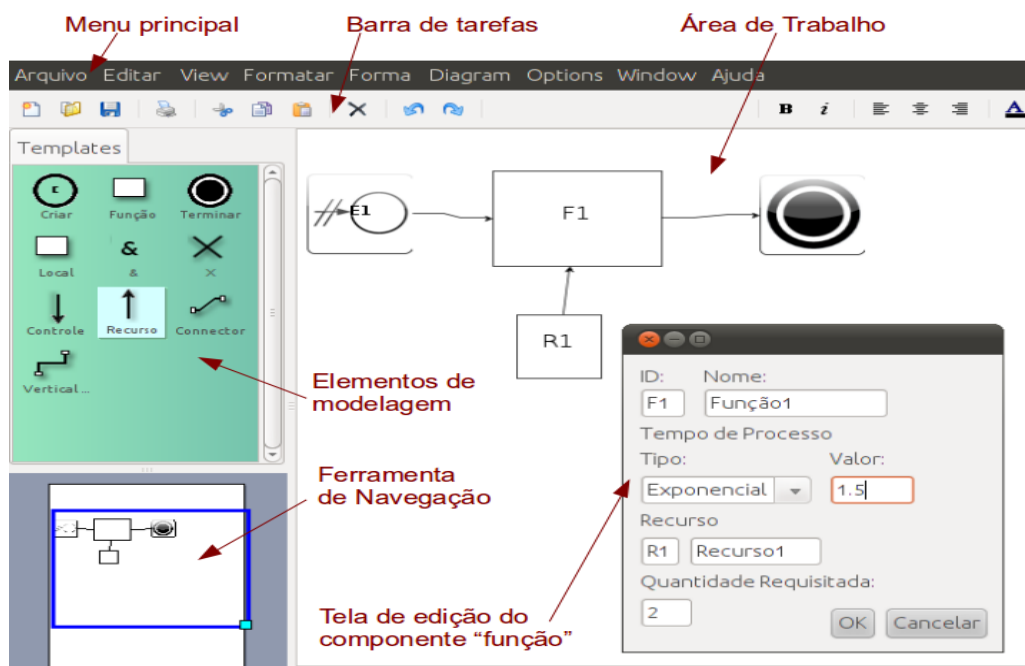


Figura 1: Interface gráfica do Ururau

O Ururau é livremente acessado em: <https://bitbucket.org/tulioap/Ururau/downloads>.

A Figura 1 também apresenta um pequeno modelo construído na área de trabalho e uma janela aberta para configurar um dos elementos do modelo. Devido à facilidade do usuário criar e configurar elementos do modelo de modo gráfico, o Ururau permite a criação de modelos de modo simples, não sendo necessário conhecimento de um LPPG ou linguagem de simulação.

Apesar dos modelos serem construídos graficamente pelos usuários, internamente ao ambiente de desenvolvimento, os modelos gráficos são convertidos em linhas de códigos, e estes são executados pelo núcleo do Ururau quando a simulação é acionada pelo usuário. Na Figura 2, que apresenta a estrutura do software, podem-se observar o núcleo e as demais partes deste software.

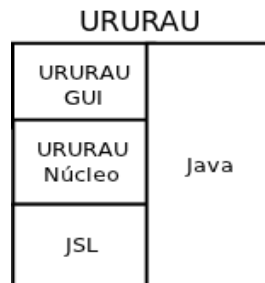


Figura 2: Estrutura do Ururau

Como pode ser visto na Figura 2, o Ururau utiliza como base a linguagem de programação Java e uma biblioteca de simulação chamada *Java Simulation Library* (JSL). Esta biblioteca oferece diversos recursos essenciais de simulação que tornam o desenvolvimento de modelos de simulação na linguagem Java mais rápidos (ROSSETTI, 2008). SILVA *et al* (2011) comparam a biblioteca JSL com um ambiente de simulação amplamente utilizado e concluíram que o JSL possui resultados de simulação e desempenho computacional compatíveis com outros softwares de simulação comercial.

O Ururau utiliza o JSL internamente para executar simulações a partir de um modelo construído graficamente, mas também permite que o usuário utilize a sintaxe desta biblioteca e do núcleo do ambiente para construir ou editar modelos de simulação usando diretamente linhas de código. Neste caso, a modelagem pode ser feita com o apoio de uma ferramenta de desenvolvimento Java, como NetBeans. A Figura 3 mostra o mesmo modelo do exemplo da Figura 1 sendo construído diretamente usando linhas de código em Java.

```

36 public static Model buildModel(){
37     // criar modelo
38     Model m = Model.createModel();
39     ProcessDescription exemplo = new ProcessDescription(m, "Modelo de Exemplo");
40     DistributionIfc E1 = new Normal(3,0.0625);// Entidade E1
41     Resource R1 = new Resource(m, 1, "Recurso1");// Recurso R1
42
43     //Função F1
44     RandomVariable F1 = new RandomVariable(m, new Exponential(1.5), "Função F1");
45     Queue Qf1 = new Queue(m, "Fila de F1");
46     Variable Rf1 = new Variable(m, 2,0, "recursos requisitados em F1");
47
48     //início da execução do modelo
49     new EntityProcessGenerator(m, exemplo, E1, E1);
50     exemplo.addProcessCommand(new Process(m, Rf1, R1, Qf1,F1));
51     exemplo.addProcessCommand(new Terminate(m, true, "Saída do sistema"));
52     return m;
53 }
    
```

Figura 3: Criação de modelo no Ururau usando linhas de código

Como pode ser visto na Figura 4, a flexibilidade de criar e modificar modelos usando a interface gráfica ou linhas de código ocorre em três níveis diferentes:

- Nível alto: recursos de ambientes de simulação (Ururau GUI);
- nível intermediário: recursos similares a linguagens de simulação (Ururau núcleo e JSL); e
- nível baixo: recursos de LPPG (JAVA).

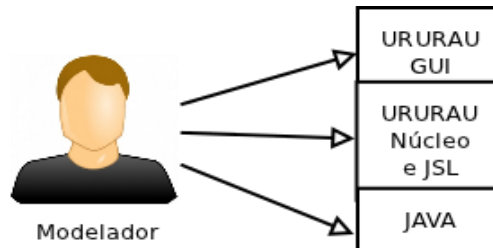


Figura 4: Possibilidades do Ururau

Por permitir a implementação nos três níveis citados, pode-se dizer que o Ururau possibilita a união, em uma única ferramenta, das vantagens dos ambientes de simulação, linguagens de simulação e LPPG.

#### 4 MODELOS DE SIMULAÇÃO COM REPRESENTAÇÕES DE DECISÃO HUMANA

De acordo com Zülch (2006), a simulação de processos industriais se concentra principalmente na modelagem de recursos (máquinas) e no fluxo de material, mas a inclusão de aspectos humanos no modelo de simulação vem ganhando importância gradativamente. Geralmente, quando os recursos humanos são modelados, apenas as tarefas operacionais estão representadas. No entanto, como resultado de conceitos modernos de produção, as decisões do trabalhador em um nível de “chão de fábrica” estão se tornando cada vez mais importantes.

Angeloni (2003) afirma que para o decisor fazer uma escolha adequada é necessário dotar de significado dados, informações e conhecimentos prévios. Para Francini (2002), a principal vantagem competitiva das empresas se baseia no capital humano, fundamentalmente, no conhecimento que os funcionários possuem. Como o conhecimento é a base para tomadas de decisão, a criação de modelos de simulação, que incluem aspectos da decisão humana baseadas em conhecimento, pode permitir uma análise mais realística do sistema em estudo.

Segundo Robinson *et al* (1998), muitos modelos de simulação incluem elementos de tomada de decisão humana que apresentam dificuldades para o modelador de simulação. Essas dificuldades são ocasionadas principalmente pela dificuldade natural de modelar decisões humanas baseadas em conhecimento e pela ausência de recursos para modelagem de decisões humanas nos ambientes de simulação de eventos discretos.

Para Robinson *et al* (2001), a maioria das ferramentas de simulação não fornecem um conjunto completo de funções que seriam úteis para modelar o comportamento humano. Além disso, de um modo geral, os recursos disponíveis nos ambientes de simulação para representar tomadas de decisão são baseadas em condições. Tais condições são incluídas no modelo usando as construções da linguagem de simulação ou simulador. Essa tarefa normalmente requer o uso de uma série de declarações 'se', 'então' e 'senão'.

A literatura trás alguns trabalhos contendo alternativas para a representação de tomadas de decisão humana em modelos de simulação:

- Robinson *et al* (1998) propõem uma abordagem utilizando um sistema especialista integrado com o modelo de simulação, onde um sistema especialista é criado para representar o tomador de decisão;
- Zülch (2006) propõe uma forma de modelagem e simulação de tomadas de decisões humanas em sistemas de manufaturas baseado em regras globais e também locais;
- Lee & Son (2008) descrevem um *framework* criado para representar o

comportamento de multidões.

Sabe-se que o homem toma suas decisões não somente pela lógica racional, mas também pela intuição, conhecimento prévio e pode ainda ser influenciado pelo seu estado emocional (QUELOPANA, 2003). As estratégias de representação de decisões humanas em simulações geralmente estão ligadas à identificação de regras lógicas (ROBINSON *et al*, 2001). Tal abordagem privilegia bastante o aspecto lógico racional das decisões, mas dá pouca ou nenhuma ênfase à intuição e ao conhecimento prévio.

Como o conhecimento dos funcionários de uma empresa pode ser considerado um dos maiores diferenciais competitivos da mesma, as abordagens de representação de decisões humanas em simulações computacionais que privilegiem o conhecimento merecem uma atenção especial (FRANCINI, 2002). Mas, como representar as decisões humanas em simulações computacionais favorecendo o aspecto do conhecimento humano? Para responder esse questionamento pode-se recorrer à inteligência artificial, mais especificamente a rede neural artificial.

### 5 REDE NEURAL ARTIFICIAL (RNA)

Segundo Novais *et al.* (2008), a RNA busca a resolução de problemas imitando o comportamento interno do cérebro humano. Dessa forma, a RNA representa de maneira artificial os neurônios biológicos e suas ligações, formando uma rede de neurônios artificiais.

Assim com uma rede neural biológica, uma rede neural artificial é capaz de guardar conhecimento. No caso das RNAs, esse conhecimento se dá através da ponderação dos pesos das conexões entre os neurônios de diferentes camadas (NOVAIS *et al.*, 2008). A Figura 5 apresenta um exemplo de RNA.

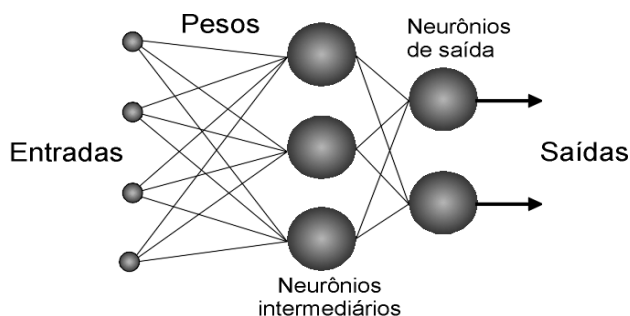


Figura 5: Exemplo de rede neural artificial (adaptado de TAFNER et al, 1995)

Para guardar conhecimento, a RNA necessita ser treinada. Um das formas de treinamento de RNA é o aprendizado supervisionado. No aprendizado supervisionado, como o próprio nome sugere, um instrutor confere o quanto a rede está próxima de uma solução aceitável, adaptando, na concepção do treinamento, os pesos entre os neurônios, de modo a prover uma menor diferença entre as saídas desejadas e a obtida (FREITAS *et al*, 2011).

Segundo Zuben (2003), uma RNA pode ser utilizada para representar decisões humanas privilegiando o conhecimento. Isso é possível devido ao modo como as RNA são treinadas (aprendizado supervisionado), onde se utilizam amostras de entradas e saídas, nas quais o vetor de entrada contém um conjunto conhecido de informações usadas como insumo para as decisões e o vetor de saída contém as respectivas decisões de uma pessoa.

### 6 MÓDULO COM INTELIGÊNCIA ARTIFICIAL

O módulo para executar a RNA onde a decisão proveniente da ação de uma pessoa pudesse ser representada durante uma simulação foi chamado de módulo inteligente. A Figura 6 ilustra a abordagem proposta, onde pode ser visto que o modelo de simulação funciona de forma acoplada ao módulo inteligente. O modelo de simulação, no caso, é desenvolvido e

executado no próprio ambiente do Ururau. Dessa forma, quando o modelo de simulação é executado, o Ururau envia os dados relacionados ao processo decisório para o módulo inteligente. Ao receber os dados, o módulo inteligente utiliza uma RNA previamente treinada para retornar a decisão.

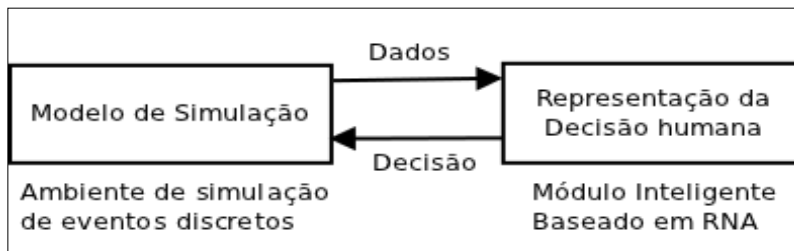


Figura 6: Abordagem proposta para simulação com representação de decisões humanas

A abordagem proposta assemelha-se ao trabalho de Robinson *et al* (1998) por demandar a criação de um sistema externo, mas diferencia-se pela estrutura e forma de comunicação entre os modelos. A abordagem proposta por Robinson *et al* (1998) prevê a criação de um controlador de modelo responsável por fazer o intermédio entre o modelo de simulação e o sistema especialista, como pode ser visto na Figura 7.

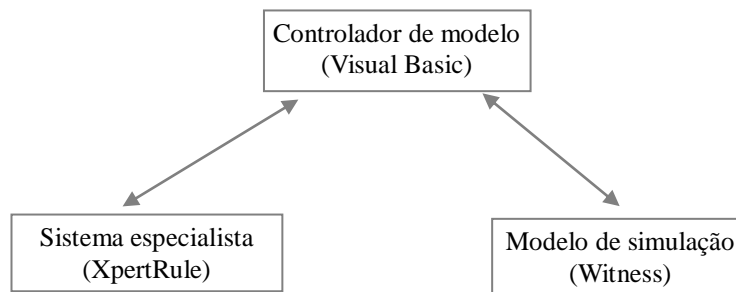


Figura 7: Comunicação entre o sistema especialista e modelo de simulação. Adaptado de ROBINSON *et al* (1998)

Devido principalmente à flexibilidade do Ururau, neste trabalho, a comunicação entre o modelo de simulação e o módulo inteligente acontece sem intermediários. Espera-se, com isso, que a construção de modelos seja mais fácil do que na abordagem proposta por Robinson *et al* (1998). Além disso, a ausência de softwares intermediários pode resultar em melhor performance computacional durante a execução da simulação. Esta questão pode ser de interesse quando se realiza uma simulação de sistemas complexos e com muitas variáveis.

## 7 AMBIENTE PARA EXECUÇÃO DO MÓDULO INTELIGENTE

O ambiente para desenvolvimento e execução do módulo inteligente foi composto pelo modelo de simulação desenvolvido em Ururau e pelo software de criação e treinamento de RNA denominado JAVANNS (FISCHER *et al*, 2001).

Optou-se pelo Ururau pela sua flexibilidade, comum aos softwares livres. Isto permitiu realizar as adaptações necessárias para que fossem implementadas diretamente no núcleo do ambiente de simulação.

O JAVANNS, desenvolvido pela Universidade de Stuttgart, foi utilizado pela sua facilidade de uso e pela possibilidade de geração de código fonte de RNA em C/C++ ou JAVA. Tal característica do JAVANNS explica, em parte, a escolha da linguagem C na programação do módulo inteligente. Além disso, sabe-se que algoritmos implementados em C/C++ possuem bom desempenho computacional.

Durante a implementação do módulo inteligente, foram seguidas as seguintes etapas:

- Adaptação do Ururau;
- criação do modelo de simulação;
- criação e treinamento da rede neural artificial; e
- criação do módulo de decisão com inteligência artificial (módulo inteligente).

As etapas estão detalhadas a seguir.

### 7.1 ETAPA 1: ADAPTAÇÃO DO URURAU

A adaptação no Ururau foi desenvolvida de modo a fornecer ao modelador a possibilidade de interligar modelos de simulação com o módulo inteligente de modo prático e transparente. Para isso, é preciso chamar uma função e passar os parâmetros adequados. Como mostra a Figura 8, a interligação do modelo de simulação e o módulo inteligente estão disponíveis no nível intermediário (Ururau núcleo e JSL).

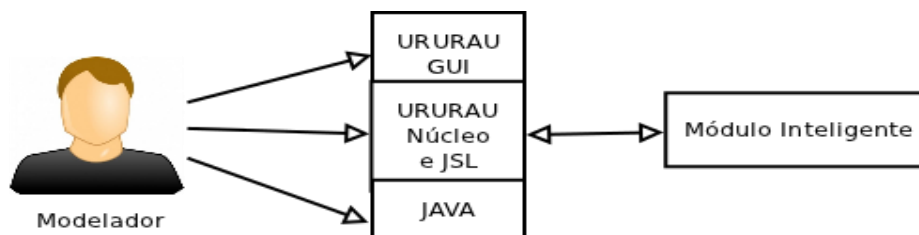


Figura 8: Interligação entre o Ururau e o módulo inteligente

Para viabilizar esta interligação, fez-se necessário adaptar o núcleo do Ururau. Foram criadas rotinas internas de comunicação, baseadas em *sockets* TCP, que possibilitaram a comunicação com o módulo inteligente, como apresentado na Figura 9. Esta implementação permite que o Ururau e o módulo inteligente sejam executados em computadores diferentes, interligados via rede *ethernet*.

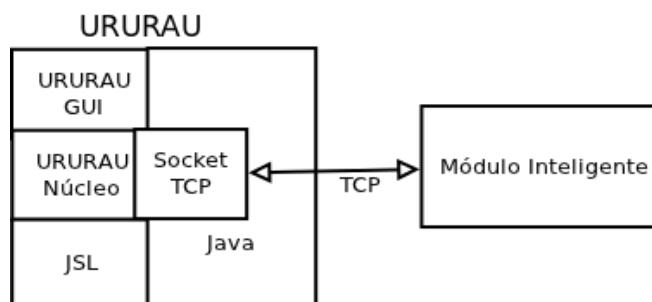


Figura 9: Adaptação do Ururau

### 7.2 ETAPA 2: CRIAÇÃO DO MODELO DE SIMULAÇÃO

Para a elaboração do modelo de simulação, foi seguida a metodologia apresentada por Banks *et al* (2010), com os seguintes passos específicos neste trabalho: Formulação e análise do problema hipotético; construção do modelo conceitual; verificação e validação; experimentação; e interpretação e análise dos resultados.

A partir da técnica IDEF-SIM (MONTEVECHI *et al*, 2010), foi possível construir o modelo conceitual do processo com um aspecto visual de fácil entendimento e lógica semelhante à utilizada na programação do modelo computacional. O modelo conceitual foi traduzido para o software Ururau para a realização das simulações computacionais.

Adicionalmente, utilizou-se a metodologia de trabalho proposta por Sargent (2011) para a verificação e validação do modelo. A validação foi realizada através do uso de dados determinísticos no modelo de teste, variações sobre os dados de entrada e uso de rotinas de



rastreamento.

Convém ressaltar que o modelo computacional foi construído após o modelo conceitual estar pronto, verificado e validado por completo. Para descrever a dinâmica dos processos, foram utilizadas funções de distribuição do tipo Normal. A função Normal pode ser utilizada sempre que a aleatoriedade for causada por várias fontes independentes, agindo de forma aditiva em torno de um ponto médio (FREITAS FILHO, 2008). Assim, a função Normal foi empregada em processos que representaram ações repetitivas, com um ponto médio bem definido, não comprometendo assim a aleatoriedade do modelo de simulação.

A Figura 10 apresenta o modelo conceitual do sistema hipotético proposto. Observe os três elementos descritos nos quadrados com “X”. Estes são os pontos de decisão do modelo onde ocorrem os desvios no fluxo das entidades. O primeiro elemento “X” (Decisão 1), mais à esquerda, representa uma decisão baseada em porcentagem, normalmente oriunda de dados históricos de um processo. O segundo elemento “X” (Decisão 2) representa uma decisão com controle oriundo de uma operação lógica tradicional para a tomada de decisão. Já o terceiro e último “X” (Decisão 3), mais à direita, é o elemento com uma decisão proveniente da RNA que busca representar melhor a ação de uma pessoa agindo no processo.

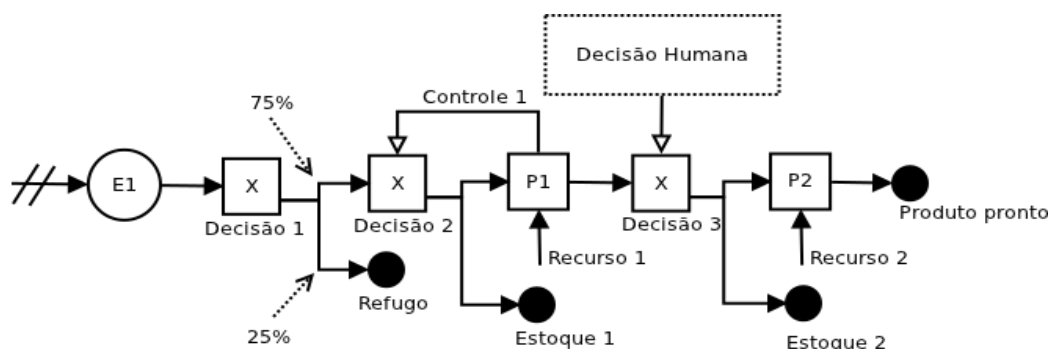


Figura 10: Modelo conceitual em IDEF-SIM com “decisão 3” usando RNA

Da forma como foi idealizado o modelo da Figura 10, podem-se executar, em um simples modelo de simulação a eventos discretos, as três possibilidades para a tomada de decisão. Assim, o modelo proposto permite a realização de experimentos com o objetivo de testar o mecanismo proposto com uma decisão mais realística utilizando RNA. Ou seja, o modelo hipotético permite confrontar o mecanismo proposto com os outros que são normalmente utilizados em decisões e é normalmente parte integrante dos softwares de desenvolvimento de modelos de simulação a eventos discretos.

Como pode ser observado na Figura 10, o fluxo inicia-se com a chegada das entidades E1, sendo em seguida desprezadas vinte e cinco por cento. As demais entidades que seguem no processo são encaminhadas para o processamento P1 ou para o estoque 1, de acordo com um controle lógico. Em seguida, as entidades que continuam seguindo o fluxo são encaminhadas para o processamento P2 ou para o estoque 2, de acordo com a decisão de um especialista. O Quadro 1 apresenta o detalhamento do modelo proposto, e deve ser analisado em conjunto com a Figura 10.

Ao término da criação do modelo conceitual, o mesmo foi implementado no Ururau, como pode ser visto na Figura 11. Nesta tarefa, recursos dos três níveis de interface do software foram usados (ver Figura 4), o que facilitou e acelerou o desenvolvimento do modelo, mesmo este tendo características especiais de representação de decisão humana.

### 7.3 ETAPA 3: CRIAÇÃO E TREINAMENTO DA REDE NEURAL ARTIFICIAL

No modelo idealizado, definiu-se que o operador tomaria suas decisões com base nos três parâmetros: Tamanho da fila P2, tempo de vida da entidade E1 e turno. Desta forma, criou-se no JAVANNS uma RNA com três neurônios na primeira camada, um para cada parâmetro de

## PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

entrada, três neurônios na camada oculta e um neurônio na camada de saída. A estrutura da RNA pode ser vista na Figura 12.

Quadro 1: Detalhamento do Modelo conceitual

Código	Significado	Descrição/Distribuição/Capacidade
E1	Entidade	<b>Distribuição:</b> Exponencial(5) minutos; <b>Atributo:</b> “Tempo de vida” Normal(6, 3) em horas. Este atributo representa o tempo entre a criação da entidade e sua entrada no sistema. Tal atributo pode ajudar a controlar, por exemplo, a validade de um insumo perecível;
P1	Processamento 1	<b>Distribuição:</b> Constante(50) minutos;
P2	Processamento 2	<b>Distribuição:</b> Normal(60; 0.2) minutos;
Recurso 1	Operador 1	<b>Capacidade:</b> 1;
Recurso 2	Operador 2	<b>Capacidade:</b> 1;
Controle 1	Controle lógico	<b>Descrição:</b> Entidades seguem para P1 Se “Fila de P1” < 2;
Refugo	Descarte de insumo	<b>Descrição:</b> Saída do sistema;
Estoque 1	Armazenagem de não processados	<b>Descrição:</b> Saída do sistema;
Estoque 2	Armazenagem de pré processados	<b>Descrição:</b> Saída do sistema;
Produto Pronto	Armazenagem de produtos prontos	<b>Descrição:</b> Saída do sistema;
Decisão humana	Representação da decisão de um especialista, baseada em RNA	<b>Descrição:</b> Para tomar essa decisão, o especialista baseia-se em três informações: <ul style="list-style-type: none"> <li>▲ Tamanho da fila em P2;</li> <li>▲ tempo de vida; e</li> <li>▲ Turno {manhã, tarde ou noite} <ul style="list-style-type: none"> <li>◦ manhã: 05:00 às 12:59;</li> <li>◦ tarde: 13:00 às 20:59;</li> <li>◦ noite: 21:00 às 04:59;</li> </ul> </li> </ul>

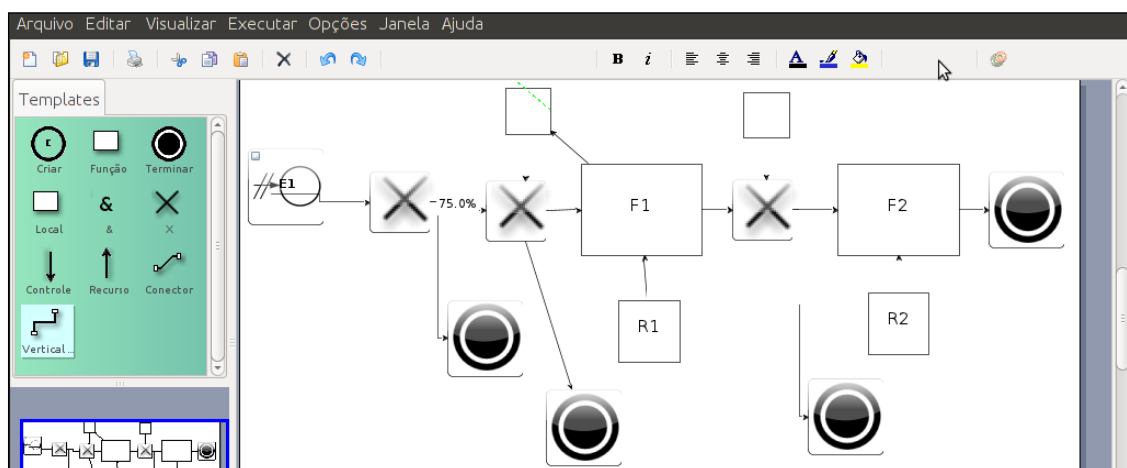


Figura 11: Modelo em Ururu

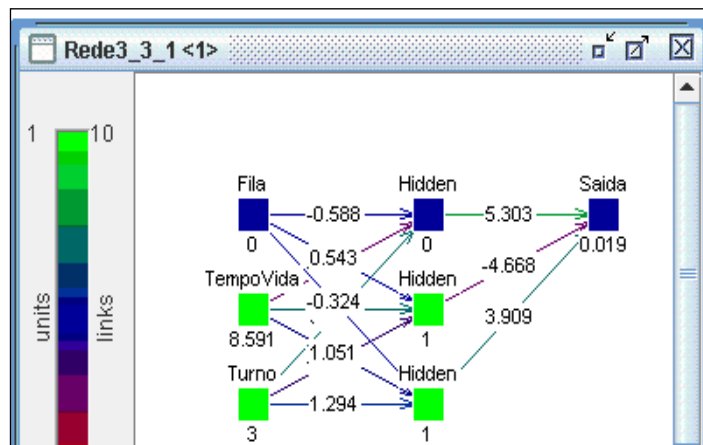


Figura 12: RNA em JAVANNS

Para realizar o treinamento da RNA, criou-se um conjunto de dados contendo diversos vetores com possíveis valores de entrada para “Fila”, “Tempo de vida” e “Turno” e as respectivas decisões esperadas.

Para elaborar os experimentos, utilizou-se o algoritmo *backpropagation* para treinamento da RNA. Foram testados alguns valores para os parâmetros *taxa de aprendizado* ( $\eta$ ) e *diferença máxima tolerada entre o valor de saída e valor desejado* ( $d_{\max}$ ).

O parâmetro taxa de aprendizado foi testado variando de 0,1 a 0,3, pois segundo Vieira & Roisenberg (1998)  $\eta$  deve ficar geralmente entre 0,1 à 0,3. Valores muito grandes para este parâmetro podem fazer o vetor gradiente ficar “saltando” de um lado para o outro na superfície de erros.

Para Diniz *et al* (2002), o valor típico de  $d_{\max}$  é 0,1, e por esse motivo esse valor foi utilizado. Testou-se também  $d_{\max}$  igual a 0,5, pois no problema em questão a saída é booleana, logo é aceitável que valores menores que 0,5 sejam considerados com 0 e valores maiores que 0,5, considerados com 1. Os testes foram efetuados buscando um erro médio quadrático menor que 0,01. A Tabela 1 apresenta o resultado dos testes.

Tabela 1: Testando parâmetros do algoritmo backpropagation

$\eta$	$d_{\max}$	Nº máximo de ciclos
0,1	0,1	2300
0,2	0,1	1360
0,3	0,1	1200
0,1	0,5	2100
0,2	0,5	1240
0,3	0,5	1180

Com base nos testes, optou-se por utilizar o algoritmo *backpropagation* com os parâmetros  $\eta = 0,3$  e  $d_{\max} = 0,1$ . A escolha da taxa de aprendizado fixada em 0,3 se deve aos melhores resultados em relação aos outros valores testados. Fixou-se a distância máxima tolerada em 0,1 para obter-se uma maior margem de confiabilidade nos experimentos, uma vez que valores próximos a 0,5 poderiam levar a RNA a atuar de modo indesejado em alguns casos.

#### 7.4 ETAPA 4: CRIAÇÃO DO MÓDULO DE DECISÃO COM INTELIGÊNCIA ARTIFICIAL

Após a criação e treinamento da RNA, o aplicativo SNNS2C (ZELL *et al*, 1996) foi usado para gerar automaticamente o código fonte em linguagem C da RNA treinada. O código gerado foi em seguida inserido em uma estrutura criada em linguagem C para controlar a comunicação com o Ururau, usando *sockets* TCP. A Figura 13 apresenta a estrutura do módulo inteligente.

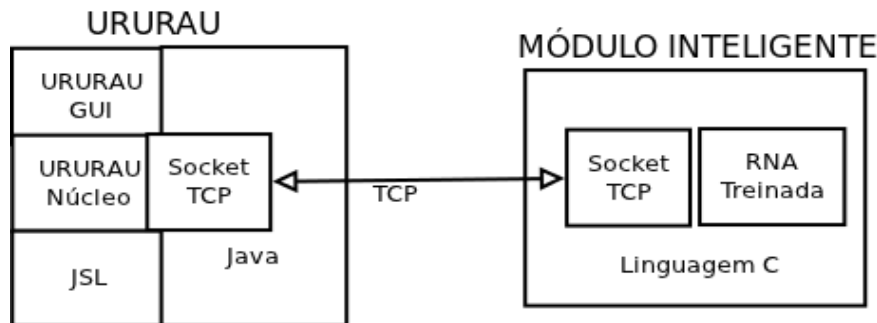


Figura 13: Estrutura do módulo inteligente

### 8 EXPERIMENTOS E ANÁLISE DOS RESULTADOS

Todos os experimentos realizados foram executados com 10 replicações com tamanho de 2880 minutos cada, em um computador com processador Dual Core 1.8 Ghz e 2GB de memória RAM.

Os experimentos foram divididos em dois grupos, de acordo com a seguinte finalidade: grupo 1: verificação da viabilidade da RNA em representar decisões humanas em modelos de simulação; e grupo 2: verificação da eficácia da RNA em representar, mais realisticamente, ações provenientes de decisões humanas em modelos de simulação.

#### 8.1 EXPERIMENTOS GRUPO 1

Foram construídos três modelos de simulação, derivados do modelo hipotético da Figura 10, para verificar a viabilidade do uso de RNA na representação de decisões proveniente da ação de uma pessoa. Tais modelos são semelhantes ao modelo hipotético, mas diferem na estratégia de representação de decisões humanas.

Como o uso de condições é atualmente a forma mais usada para representar as decisões de pessoas (ZÜLCH, 2006), optou-se por criar um conjunto de regras para representar as decisões de uma pessoa fictícia em um processo industrial.

Desta forma, criou-se um conjunto de regras que poderiam representar as decisões de um operador fictício:

- Regra 1: Se o tamanho da fila F2 for menor que 4 então a entidade E1 deverá seguir para o PROCESSAMENTO P2;
- Regra 2: Se o tempo de vida da entidade E1 for maior ou igual a 3 então a entidade E1 deverá seguir para o PROCESSAMENTO P2; e
- Regra 3: Se o turno for manhã ou tarde então a entidade E1 deverá seguir para o PROCESSAMENTO P2;

Usando o conjunto de regras descritas acima, criou-se um modelo conceitual capaz de representar as decisões humanas usando condições, identificado como “EXPERIMENTO 1”, mostrado na Figura 14. Tal modelo foi implementado e executado no Ururau. Os resultados da simulação foram registrados e também foram coletados vetores com os valores de “fila F2”, “tempo de vida E1”, “turno” e a respectiva decisão tomada durante a simulação.

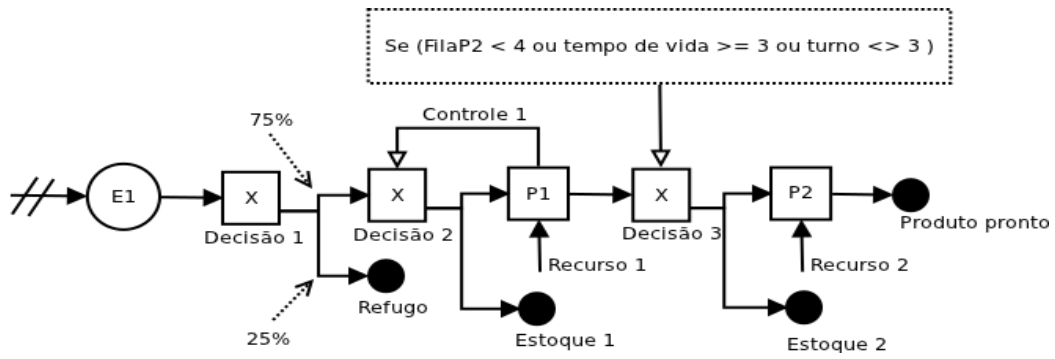


Figura 14: "Decisão 3" usando condições – EXPERIMENTO1

Utilizando-se dos resultados do EXPERIMENTO 1, calculou-se os percentuais de entidades que seguiram para P2 e Estoque 2. Em seguida, implementou-se o EXPERIMENTO2 mostrado na Figura 15, que utiliza percentuais para representar as decisões (62,1% para P2 e 37,9% para Estoque2). O EXPERIMENTO2 foi executado no Ururau e os resultados da simulação foram registrados.

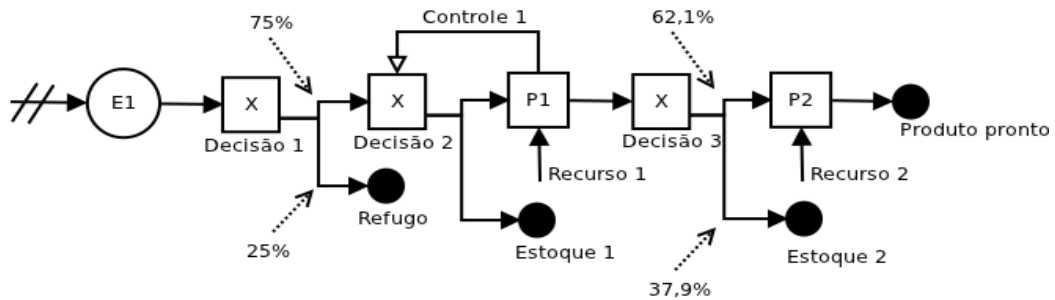


Figura 15: "Decisão 3" por percentual -EXPERIMENTO2

Finalmente, o EXPERIMENTO3 implementou o modelo conceitual apresentado na Figura 10 utilizando os vetores de decisão, coletados durante a execução do EXPERIMENTO1, contendo os dados da “fila F2”, “tempo de vida E1”, “turno” e a respectiva decisão para treinar a RNA. Usando o JAVANNNS, treinou-se a RNA e utilizou-se o SNNStoC para gerar o código fonte na RNA treinada em linguagem C. Tal código fonte foi inserido na estrutura do módulo inteligente e, finalmente, o EXPERIMENTO3 foi executado no Ururau em conjunto com o módulo inteligente.

Os resultados dos três experimentos podem ser vistos no Quadro 2.

Quadro 2: Comparação entre experimentos 1, 2 e 3

	Experimento 1	Experimento2	Experimento3
Representação da decisão humana por:	Condições	Percentual	RNA
Tempo médio das entidades na fila P2	63,8767 min	13,2576 min	63,8762 min
Quantidade média de entidades na fila P2	0,793 unidades	0,157 unidades	0,793 unidades
Decisão 3: quantidade média de entidades enviadas para “PROCESSAMENTO P2”	35,4 unidades	33,9 unidades	35,4 unidades
Decisão 3: quantidade média de entidades enviadas para “ESTOQUE 2”	21,6 unidades	23,1 unidades	21,6 unidades
Recurso R2 ocupado	73,73%	69,63 %	73,73%
Tempo computacional da simulação	0,722s	0,893 s	0,866 s

**8.2 EXPERIMENTOS GRUPO 2**

Para verificar se a RNA consegue representar mais realisticamente as tomadas de decisões, foram criados mais dois experimentos, seguindo os mesmos modelos conceituais. Optou-se, nestes experimentos, por investigar situações onde não é possível definir logicamente a decisão resultante devido à existência de ambiguidades.

Para exemplificar um conjunto de regras ambíguas, acrescentou-se uma nova regra (Regra 4) ao conjunto de regras utilizado no EXPERIMENTO1:

- Regra 1: Se o tamanho da fila F2 for menor que 4 então a entidade E1 deverá seguir para o PROCESSAMENTO P2;
- Regra 2: Se o tempo de vida da entidade E1 for maior ou igual a 3 então a entidade E1 deverá seguir para o PROCESSAMENTO P2;
- Regra 3: Se o turno for manhã ou tarde então a entidade E1 deverá seguir para o PROCESSAMENTO P2;
- Regra 4: Se o tempo de vida da entidade E1 for menor que 3 então a entidade E1 deverá seguir para o ESTOQUE 2.

O novo conjunto de regras possui ambiguidades, pois podem ocorrer, por exemplo, situações onde as condições das regras 1 e 4 sejam satisfeitas simultaneamente. Contudo, as decisões resultantes das regras 1 e 4 são diferentes, logo, configura-se uma situação ambígua. Outro exemplo são as regras 3 e 4 que também podem resultar em situações ambíguas.

Um voluntário foi convidado para usar tais regras como base para decisões e optar livremente nos casos onde acontecerem ambiguidades. Dessa forma, para a criação do EXPERIMENTO4, construiu-se um modelo de simulação no Ururau que permitia que o voluntário registrasse manualmente cada decisão tomada. Durante a execução do EXPERIMENTO4, foram coletados vetores contendo os dados da “fila F2”, “tempo de vida E1”, “turno” e a respectiva decisão.

A partir dos vetores de dados coletados no EXPERIMENTO4, uma RNA foi devidamente treinada e realizou-se o EXPERIMENTO5. Os resultados dos experimentos 4 e 5 podem ser vistos no Quadro 3.

Quadro 3: Comparação entre experimentos 4 e 5

	Experimento4	Experimento5
Representação da decisão humana por:	Voluntário	RNA
Tempo médio das entidades na fila P2	88,754 min	88,754 min
Quantidade média de entidades na fila P2	1,189 unidades	1,189 unidades
Decisão 3: quantidade média de entidades enviadas para “PROCESSAMENTO P2”	38,6 unidades	38,6 unidades
Decisão 3: quantidade média de entidades enviadas para “ESTOQUE 2”	18,4 unidades	18,4 unidades
Recurso R2 ocupado	80,40%	80,40%
Tempo computacional da simulação	2200,361 s	2,428 s

Observa-se que na comparação quantitativa apresentada no Quadro 3 os valores obtidos pelas decisões do voluntário foram realisticamente reproduzidas pela RNA.

Como os experimentos do GRUPO 2 não permitem uma abordagem usando condições, optou-se por apresentar uma comparação qualitativa entre as três formas de representação de decisões humanas abordadas nos cinco experimentos realizados, como pode ser visto no Quadro 4.

Quadro 4: Comparação qualitativa entre formas de representação de decisões humanas

	Decisão por Percentual	Decisão por Condições	Decisão por RNA
Representação de decisão humana.	Pouco realística	Realística, mas não consegue representar situações ambíguas ou excessivamente complexas	Realística
Necessita compreender a lógica racional da tomada de decisão.	Não	Sim	Não
Viável para modelar decisões humanas realísticas.	Não	Sim	Sim
Viável para modelar decisões humanas realísticas em problemas onde não é possível determinar regras lógicas usadas para a tomada de decisões.	Não	Não	Sim

## 9 CONSIDERAÇÕES FINAIS

Os resultados obtidos com os experimentos possibilitaram avaliar a abordagem proposta e demonstraram a viabilidade da aplicação de RNA para representar decisões humanas mais realísticas em modelos de simulação a eventos discretos. Ou seja, o módulo inteligente proposto e acoplado ao modelo de simulação pôde reproduzir de maneira satisfatória as decisões idealizadas como sendo proveniente da ação de uma pessoa em um sistema típico a eventos discretos.

Observou-se que o tempo computacional da simulação com decisão por RNA foi compatível com os tempos computacionais dos demais experimentos, o que confirma a viabilidade do uso de RNA para representar decisões humanas em modelos de simulação a eventos discretos.

Em relação ao Ururau, verificou-se sua grande flexibilidade. O fato da interface gráfica do software ser baseada na simbologia IDEF-SIM facilitou o aprendizado de seus recursos. Além disso, confirmou-se a utilidade do Ururau para uso acadêmico, especialmente para a experimentação de novos recursos, módulos ou algoritmos.

A comunicação proposta entre o software Ururau e o módulo inteligente usando *sockets* TCP possibilita a execução da RNA em máquinas remotas. Isto pode ser útil em situações que exijam grande esforço computacional e processamento paralelo.

### 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 BIBLIOGRÁFICAS

- ANGELONI, M.T. (2003). Elementos Intervenientes na Tomada de Decisão. *Ciência da Informação*, Brasília, V. 32, N. 1, 17-22.
- BANKS, J.; CARSON, J.S.; NELSON, B.L.; NICOL, D.M. (2010). *Discrete-Event System Simulation*. 5. Ed. New Jersey: Prentice Hall, 2010.
- CHWIF, L.; MEDINA, A.C. (2010). *Modelagem E Simulação de Eventos Discretos, Teoria e Aplicações*.

3ª Ed. São Paulo: Bravarte.

DINIZ, H.; MENDES FILHO, E.F.; CARVALHO, A.C.P.L.F. (2002). Stuttgart Neural Network Simulator: Manual de Referência Rápida. Laboratório de Inteligência Computacional da Universidade de São Paulo, São Paulo, 57 P.

FISCHER, I.; HENNECKE, F.; BANNES, C.; ZELL, A. (2001). Janns: Java Neural Network Simulator User Manual. University Stuttgart. Disponível Em <[Http://Www.Ra.Cs.Uni-Tuebingen.De/Software/Javanns/Welcome\\_E.Html](http://www.ra.cs.uni-tuebingen.de/software/javanns/welcome_e.html)>

FRANCINI, W.S. (2002). A Gestão do Conhecimento: Conectando Estratégia e Valor Para a Empresa. Rae-Eletrônica, Editora Fundação Getúlio Vargas – Escola de Administração de Empresas de São Paulo, V.1, N.2, Jul-Dez/2002. Disponível em <[Http://Rae.Fgv.Br/Rae-Eletronica/Vol1-Num2-2002](http://rae.fgv.br/rae-eletronica/vol1-num2-2002)>

FREITAS, C.S.; MARCELINO JÚNIOR, D.L.; TANAKA, L.C.; PIRES, D.L. (2011). Redes Neurais Artificiais: Introdução e Estudo de Caso. Revista Eletrônica de Sistemas de Informação e de Gestão Tecnológica, Vol. 1, No 1.

FREITAS FILHO, P. J. (2008). Introdução à Modelagem e Simulação de Sistemas: com Aplicações em Arena. 2ª Ed. Florianópolis: Visual Books.

JACOBS, P.H.M.; LANG, N.A.; VERBRAECK, A. (2002). D-Sol: A Distributed Java Based Discrete Event Simulation Architecture. In: Winter Simulation Conference, IEEE, San Diego, 793-800.

LAW, A. (2007). Simulation Modeling and Analysis. 4ª Ed. New York: Mcgraw-Hill, 768 P.

LEE, S.; SON, Y.J. (2008). Integrated Human Decision Making Model Under Belief-Desire-Intention Framework For Crowd Simulation. In: Winter Simulation Conference, IEEE, Miami, P. 886-894.

MONTEVECHI, J.A.B.; LEAL, F.; PINHO, A. F.; COSTA, R.F.S.; OLIVEIRA, M.M.; 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, IEEE, Baltimore, p. 1624-1635.

NOVAIS, A.; NOVAIS, F.; ARAÚJO, R.; NEVES, T. (2008). Redes Neurais Artificiais. Departamento de Ciência da Computação, Instituto de Matemática - Universidade Federal da Bahia, 15p. Disponível: [https://Disciplinas.Dcc.Ufba.Br/Pub/Mata64/Semestrecorrente/Redes\\_Neurais.pdf](https://disciplinas.dcc.ufba.br/pub/mata64/semestrecorrente/redes_neurais.pdf)

PEIXOTO, T. A.; RANGEL, J.J.A.; MATIAS, I.O. (2011). Ururau - Um Ambiente de Simulação a Eventos Discretos. XLIII Simpósio Brasileiro de Pesquisa Operacional, Ubatuba - SP, P. 1-2.

QUELOPANA, E.M. (2003). Conhecimento e Decisão: Um Estudo Sobre a Relação entre o Conhecimento e a Qualidade de Decisão. 155 F. Dissertação de Mestrado (Programa de Pós-Graduação em Administração) Faculdade de Economia, Administração e Contabilidade da Universidade de São Paulo - SP.

ROBINSON, S.; ALIFANTIS, T.; HURRION, R.; EDWARDS, J. S.; LADBROOK, J.; WALLER, T. (2001). Modelling and Improving Human Decision Making With Simulation. In: Winter Simulation Conference, IEEE, Arlington, P. 913-920.

ROBINSON, S.; EDWARDS, J.S., YONGFA, W. (1998). An Expert Systems Approach to Simulating the Human Decision Maker. In: Winter Simulation Conference, IEEE, Washington, p. 1541-1545.

ROSSETTI, M. D. (2008). Java Simulation Library (JSL): An Open-Source Object-Oriented Library for Discrete-Event Simulation in Java. International Journal of Simulation and Process Modelling, Vol. 4, N. 1, p. 69-87.

SARGENT, R. G. (2011). Verification and Validation of Simulation Models. In : Proceedings of the 2011 Winter Simulation Conference, IEEE, Phoenix, p. 183-198.



## PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

SILVA, D.V.C.; RANGEL, J.J.A.; PEIXOTO, T.A. (2011). Estudo Comparativo das Ferramentas JSL e Arena para Simulação de Processos Job Shop. VII Encontro Mineiro de Engenharia de Produção, São João Del Rei, p. 1-11.

TAFNER, M.; XEREZ, M.; RODRIGUES FILHO, I. (1995). Redes Neurais Artificiais : Introdução e Princípios de Neurocomputação. Blumenau : Eko.

UCHÔA JÚNIOR, F. S. (2006). Uma Análise Comparativa entre Ambientes de Simulação e Linguagens de Propósito Geral para o Desenvolvimento de Simulações. 82 F. Monografia (Programa de Graduação em Engenharia da Computação) - Escola Politécnica de Pernambuco/Universidade de Pernambuco.

VIEIRA, R. C.; ROISENBERG, M. (1998). Redes Neurais Artificiais: Um Breve Tutorial. Laboratório de Conexão e Ciências Cognitivas da Universidade Federal de Santa Catarina (Ufsc), Florianópolis, 17 P.

ZHAO, X.; VENKATESWARAN, J.; SON, Y. (2005) Modeling Human Operator Decision-Making in Manufacturing Systems Using BDI Agent Paradigm. In: Industrial Engineering Research Conference, Atlanta, p. 1-6.

ZELL, A.; MAMIER, G.; VOGT, M. (1996). Snn: Stuttgart Neural Network Simulator - User Manual. University Stuttgart. Disponível em:  
<[http://Www.Ra.Cs.Uni-Tuebingen.De/Software/Snns/Welcome\\_E.Html](http://Www.Ra.Cs.Uni-Tuebingen.De/Software/Snns/Welcome_E.Html)

ZUBEN, F.J.V. (2003). Uma Caricatura Funcional de Redes Neurais Artificiais. Learning and Nonlinear Models – Revista da Sociedade Brasileira de Redes Neurais, Vol. 1, Nº 2, p. 66-76.

ZÜLCH, G. (2006). Modelling and Simulation of Human Decision-Making in Manufacturing Systems. In: Winter Simulation Conference, IEEE, Monterey, P. 947-953.