

INFUZZY: ferramenta para desenvolvimento de aplicações de sistemas difusos

Ederson Luis Posselt¹

Rejane Frozza¹

Rolf Fredi Molz¹

Resumo: A ferramenta de modelagem de sistemas difusos, InFuzzy, foi desenvolvida com o intuito de agir como um facilitador do uso da técnica por parte de estudantes e de profissionais interessados nessa subárea da Inteligência Artificial (IA). A lógica difusa é uma excelente técnica para a solução de problemas que envolvam o uso de dados imprecisos, porém, a sua implementação e o seu entendimento não são fatos triviais. As seguintes etapas foram contempladas: estudo dos conceitos da lógica difusa, *softwares* existentes no mercado, ergonomia e usabilidade de interfaces, modelagem e desenvolvimento da ferramenta InFuzzy e homologação das suas funcionalidades. Como resultado, apresenta-se a ferramenta desenvolvida, uma interface gráfica para modelagem de aplicações de sistemas difusos, dotada de recursos de simulação e integração com outros *softwares* por meio do protocolo UDP/IP. Até o momento, muitos usuários instalaram o software InFuzzy, dentre ao quais pesquisadores e alunos de programas de mestrado e de doutorado dos seguintes estados brasileiros: Amazonas, Bahia, Ceará, Espírito Santo, Goiás, Maranhão, Minas Gerais, Pará, Paraíba, Pernambuco, Piauí, Paraná, Rio de Janeiro, Rio Grande do Norte, Rondônia, Rio Grande do Sul, Santa Catarina, São Paulo e Distrito Federal.

Palavras-chave: Lógica difusa. Ferramenta de modelagem de sistemas difusos. Ferramenta InFuzzy.

Abstract: *The modeling tool for systems fuzzy, InFuzzy, was developed with the purpose to supply the lack of free software and act as a facilitator use of technology by students and professionals interested in this sub-area of Artificial Intelligence (AI). Fuzzy logic is an excellent technique for solving problems that involve the use of inaccurate data, but its implementation and its understanding are not trivial facts. The following steps were realized: a study of the concepts of fuzzy logic, existing softwares in the market, ergonomics and usability of interfaces, modeling and development InFuzzy tool and approval of its functionality. As a result, this paper presents the developed tool, a graphical interface for modeling fuzzy systems applications, resourced simulation and integration with other software via UDP / IP protocol. At this moment, many users installed InFuzzy software. Among these, researchers and students of master's and doctoral program the following Brazilian states: Amazonas, Bahia, Ceará, Espírito Santo, Goiás, Maranhão, Minas Gerais, Pará, Paraíba, Pernambuco, Piauí, Paraná, Rio de Janeiro, Rio Grande do Norte, Rondônia, Rio Grande do Sul, Santa Catarina, São Paulo and the Federal District .*

Keywords: Fuzzy logic, modeling tool for systems fuzzy, InFuzzy tool.

¹ Programa de Pós-Graduação em Sistemas e Processos Industriais – Universidade de Santa Cruz do Sul (Unisc).
edersonlp@yahoo.com.br; {frozza, rolf}@unisc.br

1 Introdução

Lotfi Zadeh, professor de engenharia elétrica e ciência da computação da Universidade de Berkeley na Califórnia, desenvolveu, em 1965, a teoria da lógica difusa [1]. Inicialmente criticada, porém utilizada por engenheiros e cientistas da computação com o passar do tempo [2], a lógica difusa (fuzzy logic) tem como principal característica a capacidade de operar sobre dados imprecisos, mostrando-se mais adequada para tratar modos de raciocínio aproximados em vez de exatos. Em uma citação, Lotfi Zadeh menciona: “*In almost every case you can build the same product without fuzzy logic, but fuzzy is faster and cheaper.*”[3].

Os conjuntos difusos foram desenvolvidos para determinar o quanto um elemento pertence ou não a um determinado conjunto, com o uso de graus de pertinência (μ), que são valores reais no intervalo [0;1]. Assim, é possível determinar os elementos que pertencem a um grupo e seu respectivo grau de pertinência [4]. Para a aplicação da lógica difusa, é necessária a utilização de operações sobre conjuntos difusos, sendo essas operações divididas em dois grupos: Norma-T (Norma Triangular ou T-norm) e Conorma-T (Conorma Triangular ou T-conorm). As operações são aplicadas sobre o grau de pertinência (μ) dos elementos [5].

O controlador difuso é composto de três fases distintas: fuzificação, inferência e defuzificação.

- Fuzificação: esse processo deve ser realizado para cada valor de entrada e consiste no mapeamento dos dados para conjuntos difusos relevantes. Os valores de entrada são valores discretos, provenientes, geralmente, de medições ou observações. O processo de fuzificação também é responsável pela ativação das regras relevantes para uma determinada situação [6].
- Inferência: nessa etapa, são determinadas as regras que serão ativadas e combinadas, e como resultados desse processo são produzidos conjuntos difusos de saída [6].
- Defuzificação: é o processo de conversão de conjuntos difusos em uma saída discreta. A defuzificação é a etapa que exige o maior processamento, dependendo dos algoritmos escolhidos [7]. Os algoritmos mais utilizados são: centro da gravidade, primeiro, centro e último dos máximos, altura e centro da soma [8].

Atualmente, mesmo com os benefícios da utilização da lógica difusa, essa técnica é pouco disseminada nas empresas de software como auxílio em processos. Esse fato se deve, em grande parte, à questão de sua modelagem e implementação, que requer tempo, profissionais qualificados e conhecimento sobre o assunto.

O objetivo principal deste trabalho foi modelar e implementar uma ferramenta para desenvolvimento de aplicações de sistemas difusos, incentivando a solução de problemas com essa técnica e facilitando a interação com o usuário. Para desenvolver a aplicação, fez-se necessária uma pesquisa dos conceitos da lógica difusa, estudos de ergonomia e usabilidade de *software*, análise das ferramentas e dos sistemas já existentes e modelagem das funcionalidades necessárias. Este artigo apresenta os passos para desenvolvimento da ferramenta e suas funcionalidades.

O texto está organizado em seções que contemplam a evolução do trabalho. A segunda seção apresenta os softwares de modelagem de sistema difusos existentes; a terceira seção expõe as técnicas de validação de interfaces gráficas; a quarta, as características e as funcionalidades implementadas, homologação, implementação de projetos difusos da ferramenta InFuzzy. Na quinta seção, são apresentadas as conclusões.

2 Ferramentas de lógica difusa existentes no mercado

Para atingir o objetivo deste trabalho, foram estudadas diversas ferramentas, a fim de verificar pontos positivos e negativos e, então, propor melhorias no software InFuzzy. As seguintes ferramentas foram analisadas: FuzzyF, Fuzzy Logic ToolBox^{TM1}, Mathematica^{®2} Fuzzy Logic, FIDE^{TM3}, TILShell, FuzzyTECH[®], RockOn Fuzzy Tool, SciLab Fuzzy Fuzzy Tool, UNFUZZY e XFuzzy. A escolha dos softwares mencionados deve-se à disponibilidade de *download* na internet e ao seu respectivo uso em projetos publicados.

² Marca registrada.

³ Trademark.

A ferramenta *FuzzyF* foi desenvolvida por João Ricardo Bittencourt, mestre em Ciência da Computação pela Universidade do Vale do Rio dos Sinos (Unisinos). A ideia principal desse projeto foi a elaboração de um conjunto de classes em Java que implementa rotinas da lógica difusa [9].

*Fuzzy Logic ToolBox*TM é um módulo do software Matlab[®] desenvolvido pela MathWorksTM, empresa fundada em 1984 nos Estados Unidos, na cidade Natick, Massachusetts. Esse módulo oferece um conjunto de ferramentas completas para a manipulação de sistemas difusos, tanto para a construção de modelos quanto para a simulação [3].

Mathematica[®] é um produto desenvolvido pela Wolfram Research, empresa fundada em 1987 por Stephen Wolfram. Trata-se de um software de matemática onde é possível adicionar o módulo de lógica difusa [10].

FIDETM (*Fuzzy Inference Development Environment*) é uma ferramenta de sistemas difusos desenvolvida pela Apronix, empresa fundada em Tokyo, no Japão, em 1988, por um grupo de matemáticos e engenheiros. A empresa conta com filiais em outros países [11].

TILShell é uma ferramenta de criação de modelos difusos criada pela empresa Togai InfraLogic, companhia fundada pelo Dr. Masaki Togai em 1987 [12]. Esse sistema é distribuído em três versões: profissional, padrão e básica.

FuzzyTECH[®] é uma ferramenta de modelagem de sistemas difusos desenvolvida pela INFORM GmbH. Um item que diferencia esse sistema do restante são as características de ergonomia e usabilidade [13].

RockOn é um software de código aberto que implementa funcionalidades da lógica difusa. Esse software foi desenvolvido na Universidade de Kassel, na Alemanha, durante um curso de lógica difusa, pelos programadores Florian Backmann, Stefan Stützer e Stefan Lind [14].

SciLab é um software matemático distribuído gratuitamente na internet e de código aberto desde 1994. Atualmente, é dirigido pelo consórcio SciLab, fundado em maio de 2003 [15].

UNFYZZY foi desenvolvido pelo professor Óscar Germán Duarte Velasco, orientado pelo professor Gustavo Pérez, como tese de mestrado, em 1997, pela Universidade da Colômbia [16].

XFuzzy é uma ferramenta livre cuja distribuição e modificação são regidas pela licença *General Public License* (GNU). Essa ferramenta pertence aos desenvolvedores e ao Instituto de Microelectrónica de Sevilla – Centro Nacional de Microeletrônica (IMSE-CNM) [17].

A avaliação dos sistemas foi realizada mediante um processo investigativo no *site* dos desenvolvedores e nos manuais das ferramentas. Na Tabela 1, é apresentado um comparativo das principais características das ferramentas de sistemas difusos analisadas. Logicamente, algumas das ferramentas apresentam funcionalidades além das relacionadas nessa tabela, porém foram utilizadas as características fundamentais para o desenvolvimento de sistemas difusos. Os números na primeira linha representam as ferramentas na seguinte sequência: 1- FuzzyF, 2- Fuzzy Logic ToolBoxTM, 3- Mathematica[®] Fuzzy Logic, 4- FIDETM, 5- TILShell, 6- FuzzyTECH[®], 7- RockOn Fuzzy Tool, 8- SciLab Fuzzy Fuzzy Tool, 9- UNFUZZY, 10- XFuzzy e 11- InFuzzy (ferramenta desenvolvida e apresentada neste artigo).

Dentre as ferramentas pagas, o Matlab[®] e o FuzzyTECH[®] apresentaram o maior número de recursos e uma interface gráfica adequada. A ferramenta FuzzyTECH[®] leva vantagem sobre o Matlab[®] no quesito exportação de código fonte e simulação, onde é possível obter entradas externas como a porta serial.

Entre as ferramentas livres, a XFuzzy se mostrou a melhor. Essa ferramenta possui os recursos presentes nos softwares pagos como Matlab[®] e FuzzyTECH[®], além de apresentar código fonte aberto, o que permite novas implementações, sendo regida pela licença GNU.

Todavia, nenhuma das ferramentas tiveram todos os recursos implementados. A seguir, são citados alguns recursos importantes distribuídos entre as ferramentas: somente o FuzzyTECH[®] disponibiliza rotinas de relatório do projeto. Esse relatório é gerado, automaticamente, durante o processo de montagem do controlador difuso. Esse recurso é interessante para fins de controle e manutenção do sistema; o Matlab[®] não possui exportação de código fonte para linguagens de desenvolvimento; a XFuzzy não gera relatórios dos projetos implementados e também não oferece recursos de entrada de dados para simulação através de interfaces como

sockets e portas seriais; a FuzzyTECH® não traz transparência nas suas rotinas; tanto o Matlab® quanto o XFuzzy exibem, em seus manuais, fórmulas e métodos utilizados.

Tabela 1: Comparativo entre as ferramentas de modelagem de sistemas difusos.

Atributos/ Ferramentas	1	2	3	4	5	6	7	8	9	10	11
Máquina de inferência											
Mamdani	X	X	X	X			X		X	X	X
Takagi-Sugeno		X	X							X	
Próprio						X					
Fuzificação											
Gaussiana	X	X	X			X		X	X	X	X
Triangular	X	X	X	X	X	X	X	X	X	X	X
Trapezoidal	X	X	X	X	X		X	X	X	X	X
Cauchy		X	X					X	X	X	X
Sigmoide		X	X					X	X	X	X
Inclinação direita e esquerda											X
Rampa direita e esquerda											X
Discreta											X
Interface											
Gráfica		X		X		X	X		X	X	X
Linha de comando		X	X	X				X			
Defuzificação											
Centro da gravidade	X	X	X			X	X	X	X	X	X
Ponto central da área	X	X	X								
Média dos máximos	X	X	X			X	X	X	X	X	X
Primeiro dos máximos		X	X				X	X	X	X	X
Último dos máximos		X	X				X	X	X	X	X
Altura		X	X				X	X	X	X	X
Exporta Código Fonte											
C		X		X	X	X			X	X	
JAVA				X		X				X	
Assembly				X		X					
Licença											
Pago		X	X	X	X	X					
Gratuito	X						X	X	X	X	X
Código fonte aberto	X						X	X		X	
Simulação											
Possui módulo simulação		X	X	X	X	X	X			X	X
Depuração											
Possui depuração				X							X
Integração com outros sistemas											
Possui integração						X					X
Documentação											
Manual bom		X	X			X		X		X	X
Manual regular	X			X	X				X		
Manual ruim							X				
Videoaulas											X

Concluindo, após a análise de várias ferramentas, foi possível observar que todas apresentam pontos fracos e fortes, o que apontou a necessidade de uma nova ferramenta que apresentasse maior facilidade de aprendizado, direcionada principalmente para o público acadêmico e interessados na área. Todas as ferramentas analisadas exigem um pré-conhecimento conceitual avançado de lógica difusa para o seu uso. Em contrapartida, a ferramenta InFuzzy tem como um de seus pontos fortes a capacidade de elucidar o conceito e as técnicas difusas por meio da ferramenta, permitindo que usuários iniciantes modelem seus projetos em pouco tempo, ampliando o conhecimento na área. Assim sendo, o InFuzzy traz, na sua instalação, duas horas de videoaulas,

manual completo e um sistema com interface gráfica baseada em técnicas, conceitos de ergonomia e usabilidade de interfaces em software.

3 Ergonomia e usabilidade de interfaces

O sucesso de um sistema está atrelado, diretamente, com a sua ergonomia e usabilidade. É comum escutar reclamações de usuários referentes à dificuldade de uso e aprendizado de um novo sistema. Um software, ou até mesmo um eletrodoméstico, pode estar dotado das melhores e mais complexas funções, porém, se essas funcionalidades não estiverem claras e concisas, os usuários não farão uso. Conforme Cybis (2003), usabilidade é característica responsável pela eficácia e eficiência de um software, e ergonomia é a aplicação da ciência no que diz respeito a conforto, segurança e eficácia de instrumentos, máquinas e dispositivos.

Um dos conceitos mais relevantes, em se tratando de interfaces, consiste nas metáforas. As metáforas são utilizadas em interfaces para facilitar a compreensão do usuário quanto às funcionalidades de um sistema. Um exemplo de metáfora que pode ser citado é o sistema de diretórios do sistema operacional Microsoft Windows, onde é possível arrastar um determinado conteúdo para dentro de uma pasta. O usuário, ao executar essa tarefa, fica com a sensação clara de que o arquivo foi movido para a pasta, porém, o que aconteceu realmente foi a alteração da estrutura dos índices do sistema operacional [18].

A usabilidade de interfaces é um requisito para o sucesso ou a falha de um projeto de software. Conforme Nielsen (1993), a usabilidade está relacionada a cinco atributos: facilidade de aprendizado, eficiência, facilidade de lembrar, tratamento de erros e satisfação subjetiva.

Logo, desenvolvimento de software é muito mais do que simplesmente escrita de códigos. Desenvolver softwares requer planejamento e processos. Para a execução de um projeto, estão envolvidas técnicas e métodos [15].

Avaliar um software como etapa única ao final do projeto pode representar uma remodelagem do projeto como um todo. O ideal é que, durante as etapas de desenvolvimento, ocorram as validações de usabilidade [19]. A melhor forma de saber se um sistema atende aos requisitos é por meio de avaliação. Para a ergonomia e usabilidade de interfaces, isso não é diferente. Existem diversas técnicas para a avaliação da usabilidade. Nos próximos parágrafos, serão abordadas as seguintes: *Cognitive walkthrough*, heurística, GOMS e *checklist*.

Cognitive walkthrough é a técnica que tem por objetivo analisar a facilidade de aprendizado de um software, baseando-se nas ações necessárias para realizar uma atividade [19]. Para a execução dessa técnica, são coletadas informações referentes ao nível do usuário, às funções do sistema e à sequência de etapas necessárias para executar as funções. Com base nos dados coletados, avaliadores fazem uma crítica em relação à usabilidade [20]. Como vantagens dessa técnica, tem-se: ênfase nos problemas do usuário, ausência de protótipos e usuários durante a avaliação. A desvantagem dessa técnica está no fato de ser muito trabalhosa e não ser aplicável a qualquer tipo de sistema [21].

A avaliação heurística foi desenvolvida por Jacob Nielsen e Rolf Molich. Essa técnica consiste em avaliar o software com o usuário através de um conjunto de heurísticas. Os dez princípios definidos são: visibilidade do *status* do sistema, compatibilidade entre o sistema e o mundo real, liberdade e controle do usuário, consistência e padrões, prevenção contra erros, reconhecimento em lugar de lembrança, flexibilidade e eficiência de uso, projeto minimalista e estético, auxiliar os usuários a reconhecer, diagnosticar e recuperar-se de erros, ajuda e documentação. Essa técnica exige a participação de vários avaliadores, que, ao final, reúnem-se e consolidam um relatório sobre os problemas encontrados. A vantagem desse método está no fato de qualquer nível de avaliador estar habilitado a executar a checagem do sistema, porém, a qualidade do resultado pode ficar atrelada à capacidade do avaliador [21].

GOMS (*Gols, Operators, Methods, Seletion Rules*) é o modelo de avaliação elaborado por Card, Moran e Newell (1983). Essa técnica se baseia nos passos necessários para realizar uma determinada tarefa no sistema. Depois de descritos o objetivo, os métodos, os operadores e as regras, é possível extrair informações de desempenho sem que o usuário precise realizar os procedimentos [20]. Existem variações da técnica de GOMS, entre as quais: CMNGOMS, KLM, NGOMSL e COM-GOMS. Como vantagem desse método, pode-se citar a possibilidade de execução sem a presença de usuários, e como desvantagem, a dificuldade de mapear o conjunto de ações em sistemas flexíveis [21].

Checklists são listas de verificação de ergonomia de software. Por meio dessas listas, profissionais como programadores e analistas podem executar a validação da interface sem a necessidade de especialistas. O sucesso desse tipo de avaliação está mais relacionado à qualidade da lista de verificação do que ao avaliador. Os resultados das avaliações costumam ser mais regulares, uma vez que os profissionais seguem a lista de verificação [22].

A técnica de *checklist* foi adotada na avaliação do software InFuzzy. Essa escolha se deve à facilidade de aplicação e execução por programadores e analistas. É importante salientar que existem listas de verificação elaboradas disponíveis de forma gratuita. O *checklist* é aplicado durante o processo de desenvolvimento, e não somente ao final do projeto. As questões de verificação aplicadas foram extraídas do software Ergolist [23]. Após a aplicação da técnica, observou-se que 66% das questões aplicadas são atendidas pelo sistema, 12% não são atendidas e 22% das questões não são aplicáveis ao software Infuzzy. Computando-se apenas os itens aplicáveis, verificou-se que 85% das questões de ergonomia são atendidas e apenas 15% não são atendidas. Considera-se que a interface está, em sua maioria, de acordo com as normas de usabilidade e ergonomia, proporcionando facilidade de uso e aprendizado ao usuário.

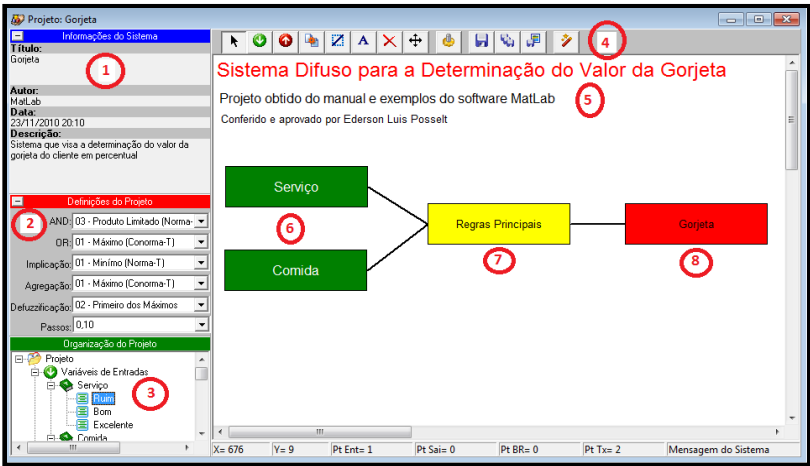
4 Ferramenta INFUZZY

A ferramenta InFuzzy foi desenvolvida em Delphi para a plataforma Microsoft nas versões XP, Vista e Seven. Esse software também funciona no sistema operacional Linux, executado pelo software *Wine* (sistema capaz de executar aplicações Windows em sistemas operacionais Linux).

A construção dos modelos difusos ocorre por meio de interface gráfica. A determinação das entradas, saídas e blocos de regras são realizados por operações de *drag and drop* (arrastar e soltar). Esses componentes são ligados por meio de linhas. A união de todos os componentes na tela dá origem ao controlador difuso modelado pelo usuário.

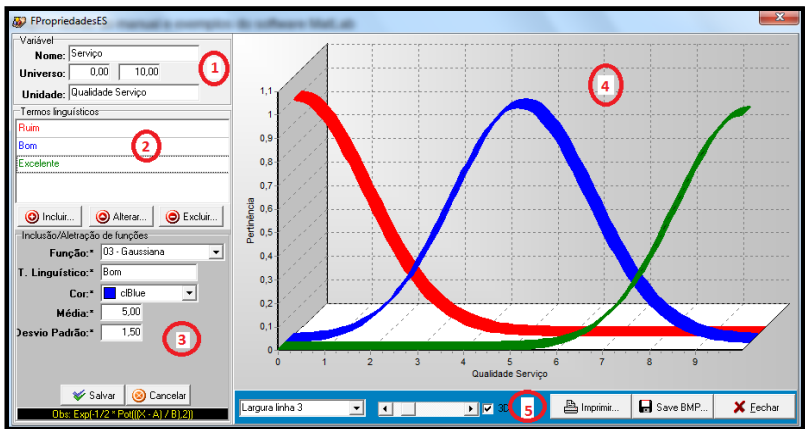
A Figura 1 apresenta a tela de construção de um projeto, sendo a tela mais importante do sistema, pois é através dessa que ocorrem todas as manipulações sobre variáveis de entrada, saída, bloco de regras e mensagens. Observando-se essa figura, há a seguinte estrutura: 1 (identificação do projeto), 2 (definição das funções utilizadas no modelo difuso), 3 (estrutura do projeto difuso em formato de árvore), 4 (barra de ferramentas), 5 (adição de texto juntamente ao projeto modelado), e, por último, os itens 6, 7 e 8 representam a estrutura do projeto modelado, com duas variáveis de entrada, um bloco de regras e uma saída.

Figura 1: Tela de controle sobre o projeto difuso.



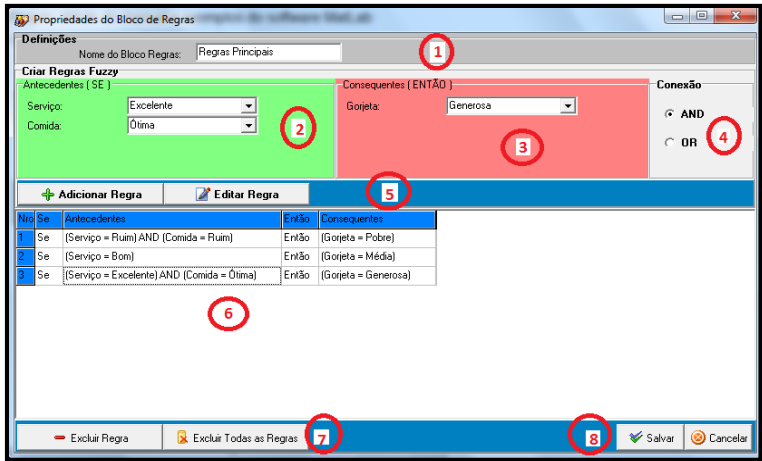
A Figura 2 apresenta a tela de configuração das variáveis de entrada e saída. A tela está dividida da seguinte forma: 1 (definições da variável e universo), 2 (termos linguísticos), 3 (inclusão e edição de termos linguísticos) e 4 (representação gráfica dos termos linguísticos).

Figura 2: Tela de edição de variáveis de entrada e saída.



A Figura 3 apresenta a tela de definição das regras, sendo essa dividida da seguinte forma: 1 (nome do bloco de regras), 2 (definição dos termos linguísticos para cada variável de entrada), 3 (definição do termo linguístico para cada variável de saída), 4 (escolha do operador de conexão entre os antecedentes), 5 (adição e edição de regras), 6 (listas das regras definidas), 7 e 8 (procedimentos de exclusão e pertinência das regras criadas).

Figura 3: Tela de definição das regras.



A Tabela 2 apresenta as possibilidades de configuração da ferramenta para o desenvolvimento de um sistema difuso. As funções de pertinência são aplicadas aos termos linguísticos de cada variável, seja ela de entrada ou saída. As relações entre as regras podem ser estabelecidas pelo operador OR ou AND, sendo que para cada um desses operadores é possível determinar uma função a ser aplicada. A implicação é a função destinada a determinar a forma como o resultado dos antecedentes ocorre sobre os consequentes. Na etapa de agregação é possível selecionar uma função para determinar como o resultado de cada regra influencia o resultado de pertinência da variável de saída. A última configuração é referente ao número de passos, essa configuração determina a precisão dos resultados de defuzzificação, pois é ela quem informa o deslocamento dentro do universo.

Tabela 2: Tabela com os opcionais de configuração do sistema difuso.

Configuração	Opções
Funções de pertinência	Triangular, Trapézio, Gaussiana, <i>Cauchy</i> , Sigmóide, Rampa Direita e Esquerda, Inclinação Direita e Esquerda, Discreto
AND	Mínimo (Norma-T), Produto Algébrico (Norma-T), Produto Limitado (Norma-T)
OR	Máximo (Conorma-T), Soma Algébrica (Conorma-T), Soma Limitada (Conorma-T)
Implicação	Mínimo (Norma-T)
Agregação	Máximo (Conorma-T), Soma Algébrica (Conorma-T), Soma Limitada (Conorma-T)
Defuzificação	Centro da Gravidade, Primeiro dos Máximos, Centro dos Máximos, Último dos Máximos, Média dos Máximos, Altura
Passo	0,01; 0,10; 0,50; 1,00; 2,00; 5,00; 10,00

Um dos maiores diferenciais entre o software InFuzzy e as demais ferramentas analisadas está na possibilidade de integração do InFuzzy com outros softwares por meio do protocolo UDP/IP. Mediante essa funcionalidade, as rotinas do software InFuzzy podem ser executadas de forma remota. O protocolo de comunicação é baseado no protocolo *National Marine Electronics Association* (NMEA), utilizado nos equipamentos de Sistema de Posicionamento Global (GPS), onde os dados são transmitidos em ASCII. Para facilitar o entendimento dessa integração, foi desenvolvido um software que acompanha a instalação do sistema InFuzzy, chamado de InFuzzy UDP Tool. Com esse software, é possível simular todas as rotinas de integração implementadas. Juntamente com o instalador do sistema, são disponibilizados o manual e videoaulas para realizar a integração.

4.1 Validação da ferramenta InFuzzy

Para realizar a homologação da ferramenta, foi utilizada a seguinte estratégia. Primeiro, foi escolhida uma ferramenta de sistemas difusos para a comparação de resultados. A ferramenta escolhida foi o Matlab®, ToolBox Fuzzy. A escolha desse software se deve ao reconhecimento desse fabricante no mundo e à qualidade e à confiabilidade apresentadas por seus produtos. No segundo momento, foi escolhida uma aplicação de demonstração presente na instalação do Matlab® versão R2009a, chamada Tipper. Essa aplicação tem como objetivo definir o percentual de gorjeta em relação à qualidade da comida e do serviço.

Para demonstrar a confiabilidade no projeto, foi criada uma lista de entradas que foi simulada em ambos os softwares. Ao longo dessa comparação de resultados, foram realizadas alterações no projeto nas duas ferramentas (Matlab® e InFuzzy), a fim de cobrir o máximo de seus recursos. Todos os resultados foram conferidos no software Matlab® e coincidiram. A ferramenta InFuzzy apresenta uma vantagem em relação ao software Matlab®, em que é permitida a criação de uma lista de valores a serem testados, e esses valores ficam salvos no projeto. Isso se torna mais importante na medida em que são realizados vários testes com configurações variadas.

Além da comparação direta com o software Matlab®, foram selecionados projetos extraídos de trabalhos de graduação, dissertações, teses e publicações na área, a fim de verificar as principais necessidades em relação a projetos difusos. A seguir, apresenta-se uma lista de alguns projetos implementados e validados:

- Uso da Lógica Fuzzy para a otimização de sistemas energéticos baseados em energia limpa [24].
- Sistema especialista em Lógica Fuzzy para o controle, gerenciamento e manutenção da qualidade [25].
- Modelo Fuzzy para estimar a durabilidade dos transformadores de alta tensão dos energizadores de cerca [26].
- Sistema de um controlador Fuzzy baseado no Labview [27].

5 Conclusões

A lógica difusa é uma técnica capaz de operar sobre dados imprecisos, mostrando-se mais adequada para tratar modos de raciocínio aproximados em vez de exatos. Atualmente, essa técnica é pouco difundida na indústria e em empresas de desenvolvimento de software. Nesse trabalho, realizaram-se uma pesquisa e uma comparação entre várias ferramentas existentes no mercado, com a finalidade de divulgação e base para o desenvolvimento do InFuzzy. A carência observada por meio da análise das ferramentas existentes, principalmente no que diz respeito à ergonomia e à usabilidade de software e documentação, possibilitou o desenvolvimento de um software gratuito, com interação gráfica de fácil manuseio, documentação completa, capacidade de realizar simulações e integração com outros sistemas mediante o protocolo UDP/IP.

Após a concretização de várias etapas, obteve-se como resultado um software com as seguintes características:

- A instalação do software é executada de forma rápida e simples, método já conhecido pela maioria dos usuários de computadores. Todos os arquivos necessários para o funcionamento e o aprendizado encontram-se no instalador, sendo eles: software InFuzzy, DLLs (*Dynamic-link library*) necessárias, manual do software, exemplos de projetos modelados, videoaulas e aplicativo para desinstalar o software.
- Interface gráfica amigável de fácil aprendizado, homologada pela técnica de *checklist*.
- Módulo de simulação com *trace* (passo a passo), permite que a cada simulação sejam acompanhados os procedimentos realizados para a obtenção dos resultados, além da visualização do processo de defuzzificação por meio de um gráfico.
- Módulo de integração com outros softwares por meio do protocolo UDP/IP. Esse módulo é um diferencial em relação aos softwares de modelagem analisados neste trabalho. Esse módulo permite ao usuário testar, de forma ágil e fácil, a lógica difusa para uma determinada aplicação com o mínimo de esforço.
- As videoaulas, que totalizam um tempo superior a 105 minutos, demonstram, de forma detalhada, o uso da ferramenta InFuzzy na seguinte organização: todos os módulos do sistema (definição de variáveis, termos linguísticos, regras, simulações), desenvolvimento de um sistema difuso completo, desenvolvimento de um sistema em hardware integrado com o software InFuzzy e o uso do software InFuzzy UDP Tool.

A ferramenta InFuzzy, desenvolvida neste projeto, resultou em um software capaz de alterar o cenário atual, aproximando a teoria da prática, gerando resultados ao usuário em um tempo curto, capaz de validar a técnica de lógica difusa em uma determinada situação. O *download* da ferramenta InFuzzy pode ser realizado por meio do *site* do Programa de Pós-Graduação em Sistemas e Processos Industriais – Unisc. (<http://www.unisc.br/portal/pt/cursos/mestrado/mestrado-em-sistemas-e-processos-industriais/programa.html>).

Referências

- [1] ZADEH, L. A. BISC: The Berkeley Initiative in Soft Computing. Disponível em: <<http://www.cs.berkeley.edu/~zadeh/>>. Acesso em: 10 nov. 2009.
- [2] COSTA, A. et al. LÓGICA FUZZY: CONCEITOS E APLICAÇÕES. Disponível em: <http://www.inf.unisinos.br/~cazella/dss/fuzzy_relatorio.pdf>. Acesso em: 10 Jun. 2009.
- [3] MATHWORKS. Fuzzy Logic Toolbox: Users Guide. Disponível em: <<http://www.mathworks.com/>>. Acesso em: 01 ago. 2009.

- [4] REZENDE, S. O. Sistemas inteligentes Fundamentos e Aplicações. In: ALMEIDA, P. E. M.; EVSUKOFF, A. G. de (Org.) Sistemas Fuzzy. São Paulo: Manole, 2003. p.169-201.
- [5] CALDEIRA, A. M. et al. Inteligência Computacional Aplicada à Administração, Economia e Engenharia em Matlab. São Paulo: Thomson, 2007. 370 p.
- [6] TANSCHKEIT, R.. Sistemas fuzzy. In: VI Simpósio Brasileiro de Automação Inteligente, 2003, Bauru, SP. Anais de Minicursos do VI SBAI, 2003. p. 35.
- [7] WEBER, L.; KLEIN, P. A. T. Aplicação da Lógica Fuzzy em Software e Hardware. Canoas: Ulbra, 2003. 110 p.
- [8] DRIANKOV, D.; HELLENDROORN, H.; REINFRANK, M. An Introduction to Fuzzy Control, Second Edition. Berlin: Springer, 1996.
- [9] BITTENCOURT, J. R.; OSÓRIO, F. FuzzyF - Fuzzy Logic Framework: Uma Solução Software Livre para o Desenvolvimento, Ensino e Pesquisa de Aplicações de Inteligência Artificial Multiplataforma. In: WORKSHOP SOBRE SOFTWARE LIVRE, 3., 2002, Porto Alegre. Anais... . Porto Alegre, RS: Sociedade Brasileira de Computação, 2002. p. 58 – 61
- [10] WOLFRAM. MATHEMATICA FUZZY LOGIC, The most flexible environment for exploring fuzzy systems. Wolfram Research, 2003. Disponível em: <<http://www.wolfram.com>>. Acessado em: Ago de 2009.
- [11] ZHU, D. FIDE -Fuzzy Inference Development Environment An Intelligent Software Tool for Mutli-Sensor Data Fusion. Apronix Application Note, 1997.
- [12] TOGAI INFRALOGIC. The World's Source For Fuzzy Logic Solutions. Disponível em: <<http://www.ortech-engr.com/fuzzy/togai.html>>. Acessado em: Nov de 2009.
- [13] FUZZYTECH. Fuzzy Logic Advantage in Your Design. Disponível em: <www.fuzzytech.com>. Acessado em: Nov de 2009.
- [14] ROCKON FUZZY TOOL, The Best Free Fuzzy Control And Simulation Tool. Disponível em: <<http://www.timtomtam.de/rockonfuzzy/>>. Acessado em: Nov de 2009.
- [15] ROSSON, M. B.; CARROLL J. M. Usability Engeneering – Scenario-based Development of Human-Computer Interaction Desing. San Diego: Academic Press, 2002.
- [16] VELASCO, O. G. D. Sistema de lógica difusa. Disponível em: <<http://www.ing.unal.edu.co/~ogduarte/softwareDetallado.htm>> Acessado em: Dez de 2009.
- [17] IMSE-CNM. Fuzzy Logic Design Tools. Manual do software. 2003.
- [18] ERICKSON, T. D. Working with interface metaphors. In Human-Computer interaction: Toward the Year 2000, R. M. Baecker, et al. San Francisco: Morgan Kaufmann Publishers, 1995, p. 147-151.
- [19] DIX, A. et al. Human-Computer Interaction. 2ª Ed. Harlow: Prentice Hall, 2001.
- [20] PREECE, J.; ROGERS, Y.; SHARP, H. Interaction design: Beyond human–computer interaction. USA: Jhon Wiley & Sons, 2002.
- [21] LUNARDI, A. Uma estratégia para a avaliação de usabilidade com suporte de um ambiente cooperativo. 2004. Graduação (Graduado) - Curso de Ciência da Computação, Departamento de Informática, UNISC, Santa Cruz do Sul, RS, 2004.

- [22] CYBIS, W. A. Engenharia de Usabilidade: uma abordagem ergonômica. Laboratório de Utilizabilidade de Informática. Florianópolis, 2003
- [23] ERGOLIST. Software para avaliação de interfaces. Disponível em: <<http://www.labiutil.inf.ufsc.br/ergolist/quest.htm>>. Acessado em: Nov de 2009.
- [24] POSSELT, E. L.; MOLZ, R. F.; HORN, F. Uso de Lógica Fuzzy para a Otimização de Sistemas Energéticos Baseados em Energia Limpa. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, XXIX, 2009, Salvador, BA.
- [25] BILOBROVEC, Marcelo. Sistema Especialista em Lógica Fuzzy Para o Controle, Gerenciamento e Manutenção da Qualidade em Processo de Aeração de Grãos. 2005. 74 f. Dissertação (3) - Curso de Engenharia de Produção, Departamento de Pós-graduação, Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2005.
- [26] HAAS, Alci Antônio. Um modelo fuzzy para estimar a durabilidade dos transformadores de alta tensão dos energizadores de cerca. 2010. 134 f. Dissertação (Mestrado) - Curso de Pós-graduação em Sistema e Processos Industriais, Unisc, Santa Cruz do Sul, 2010.
- [27] JIN, Mou-lin; HO, Ming-chun. Labview-based fuzzy controller design of a lighting control system. Journal Of Marine Science And Technology, [Osaka], v. 17, n. 2, p. 116-121, 2009. Mensal.