

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

**Identificação Automática de Problemas de Usabilidade
em Interfaces de Sistemas Web através de Lógica
*Fuzzy***

Gisele Alves Santana¹

Centro de Pesquisa em Controle e Automação (CIPECA), UTFPR, Cornélio Procópio, PR

Diony José de Almeida²

Centro de Pesquisa em Controle e Automação (CIPECA), UTFPR, Cornélio Procópio, PR

Vladimir Aparecido Freire Junior³

Centro de Pesquisa em Controle e Automação (CIPECA), UTFPR, Cornélio Procópio, PR,
Departamento de Tecnologia em Manutenção Industrial, UTFPR, Guarapuava, PR

Rodrigo Rodrigues Sumar⁴

Centro de Pesquisa em Controle e Automação (CIPECA), UTFPR, Cornélio Procópio, PR

Resumo. Recentemente, alguns sistemas estão sendo transferidos para a plataforma web. Muitos serviços e aplicações, incluindo sistemas de automação, são desenvolvidos com interfaces baseadas na Internet. A usabilidade é a principal característica de uma interface e está associada com as funcionalidades de um sistema. Ela descreve o quão bem um produto pode ser utilizado para os fins propostos por seus usuários com eficácia, eficiência e satisfação. Este artigo apresenta a aplicação de técnicas de Lógica *Fuzzy* na identificação automática de problemas de usabilidade na interface de um sistema web. Os resultados indicam que esta abordagem é satisfatória em relação à identificação automática de possíveis problemas de usabilidade na interface de um sistema web.

Palavras-chave. Avaliação de Usabilidade, Sistemas Web, Modelo de Tarefas, Análise de Log, Lógica *Fuzzy*.

1 Introdução

Usabilidade é um termo usado para definir a facilidade com que as pessoas podem usar um sistema, a fim de realizar uma tarefa específica [6]. Problemas de usabilidade

¹ giselesantana@utfpr.edu.br.

² diony.a@gmail.com

³ vladimirjunior@utfpr.edu.br

⁴ sumar@utfpr.edu.br

ocorrem quando um usuário encontra dificuldades para realizar uma tarefa com uma interface. Esses problemas podem ter origens variadas e ocasionar perda de dados, diminuição da produtividade, chegando à rejeição total do sistema.

Os sistemas de automação atuais podem possuir módulos voltados para a Internet. Através desses módulos, os navegadores podem disponibilizar interfaces gráficas nas quais o usuário interage remotamente com o processo industrial [2,3]. Desse modo, é notória a necessidade de sistemas web fáceis de serem compreendidos e bem projetados.

Muitas pesquisas estão sendo desenvolvidas na área de avaliação remota de usabilidade. Dentre essas, destaca-se o estudo de [2] que demonstram o uso de um suporte automático para avaliação da usabilidade. Os resultados mostram que com a ajuda de automação, mais problemas de usabilidade podem ser detectados, reforçando a viabilidade e eficácia da avaliação de usabilidade automatizada de um sistema.

No trabalho de [1] é utilizada inteligência computacional introduzindo lógica *Fuzzy* ao cenário da avaliação da usabilidade. Esse trabalho desenvolve um modelo de medida geral de usabilidade utilizando a teoria dos conjuntos *Fuzzy*.

Já o ambiente chamado Ergo-Monitor [5] objetiva o monitoramento da usabilidade de sistemas web por meio da coleta seletiva e da análise de dados dos arquivos de *log* referentes às interações entre usuário e sistema web.

No trabalho desenvolvido por [7], os problemas de usabilidade são identificados automaticamente por meio de técnicas de Reconhecimento de Padrões, utilizando Redes Neurais Artificiais para a classificação dos problemas identificados.

A abordagem descrita neste artigo é semelhante à de [7] em termos de filtragem dos arquivos de *log* e definição de um Modelo de Tarefas. No entanto, este trabalho não quantifica a usabilidade de um sistema, mas identifica automaticamente se um sistema web apresenta potenciais problemas de usabilidade. A identificação e classificação dos potenciais problemas de usabilidade é realizada através da aplicação de lógica *fuzzy*.

Este artigo está organizado da seguinte maneira: na Seção 2 é realizado um estudo sobre a avaliação de usabilidade. Na Seção 3 é descrito como os arquivos de *log* são coletados. Na Seção 4 é apresentado um modelo de tarefas, que descreve a interação do usuário com o sistema web. Na seção 5 é descrita uma metodologia para filtragem dos arquivos de *log*. Na seção 6 é apresentado o sistema de inferência *fuzzy* e os resultados obtidos. Na Seção 7 são apresentadas as conclusões.

2 Avaliação de Usabilidade

A técnica empregada neste trabalho é a análise de *log*, que analisa as interações do usuário registradas em arquivos de *log* gerados durante a utilização do sistema. É um método de avaliação que, além de apontar problemas de usabilidade, pode ajudar a compreender o comportamento dos usuários em relação à interface do sistema [6].

Para este trabalho, foi desenvolvido um sistema web para fins de testes. O sistema web representa uma agência fictícia de empregos e disponibiliza serviços referentes ao cadastro de currículos de usuários em diferentes áreas de atuação. A tarefa “Cadastrar currículo” é analisada, partindo do princípio que se o sistema proporciona uma boa usabilidade, o usuário não cometerá erros graves ou nenhum erro, e precisará de pouca ou nenhuma ajuda para a execução da tarefa.

3 Coleta dos Arquivos de Log

Este trabalho analisa os arquivos de *log* gerados pelo servidor Apache. Esse servidor registra uma entrada no arquivo de *log* para todo acesso que recebe. O formato de cada entrada de um arquivo de *log* gerado pelo servidor Apache é o seguinte: 127.0.0.1 - - [13/Dec/2011:15:09:57 -0200] "GET /index.html HTTP/1.1" 304

Na Tabela 1 é descrito o significado de alguns itens presentes no *log* do servidor Apache.

Tabela 1: Campos de Arquivo de *log* do servidor Apache.

Endereço do IP	127.0.0.1
Nome do cliente (- se anônimo)	-
Data da solicitação	13/Dec/2011
Horário da solicitação	17:55:58
Documento solicitado	index.html

4 Modelo de Tarefas

A ideia subjacente ao Modelo de Tarefas (MT) é a modelagem das tarefas executadas pelo usuário na interação com um sistema web e posterior análise do comportamento do usuário perante a estas tarefas.

Para o contexto deste trabalho, uma tarefa foi definida como uma sequência de passos que o usuário deve realizar para atingir um determinado objetivo. Cada um desses passos são acessos a uma página web e identificados por URLs.

Formalmente pode-se definir uma tarefa k pela seguinte quádrupla: $tk = \{S, po, pf, T\}$, onde S é um conjunto, de tamanho n , de páginas web que devem ser visualizadas pelo usuário para cada tarefa; po é a página web inicial; pf é a página final que identifica o sucesso da tarefa e, finalmente, $T : S \rightarrow S$ é a função de transição que identifica a passagem de uma página web pi para outra página web pj .

Na página inicial do sistema, o usuário tem algumas possíveis opções para cadastrar seu currículo. Assim, pode-se afirmar que o usuário que tem o objetivo de cadastrar seu currículo no sistema web tem como caminho o acesso aos arquivos: "index.html", "cadastro.jsp" e finalmente "sucesso.html". No entanto, ao defrontar-se com um problema, o usuário tem a opção de consultar a página de ajuda (ajuda.html). Se o problema persistir, o mesmo será direcionado para a página de erro (erro.jsp). O usuário somente terá êxito em seu objetivo se conseguir atingir a página de sucesso.

Dessa maneira, o MT para a tarefa "Cadastrar currículo" é definido como:

- $S = \{ \text{index.html, cadastro.jsp, ajuda.html, erro.jsp, sucesso.html} \}$
- $po = \{ \text{index.html} \}$
- $pf = \{ \text{sucesso.html} \}$

5 Filtragem dos Arquivos de Log

É possível representar uma entrada no arquivo de *log* por $li = \{ui, d, h, c, ri\}$, onde $ui \in U, ri \in R$; d é a data do acesso; h é o horário do acesso; e c o código de status.

Quando um usuário acessa uma página, ele visualiza um conjunto de recursos. A

filtragem do arquivo de *log* é realizada de forma que cada entrada não represente mais o acesso a um recurso, mas o acesso à uma página. Assim, uma entrada no arquivo de *log* filtrado é representado por $lfi = \{ui, d, h, c, pi\}$. O arquivo *log* filtrado é constituído somente pelas páginas identificadas nos dois arquivos. O Quadro 1 ilustra um trecho de um arquivo de *log* filtrado.

[04/Dec/2012:15:09:57:10] /index.html	Acesso com Sucesso
[04/Dec/2012:15:09:58:30] /cadastro.jsp	
[04/Dec/2012:15:10:02:30] /sucesso.html	
[04/Dec/2012:15:10:30:10] /index.html	Acesso com sucesso + Acesso a erro
[04/Dec/2012:15:10:31:30] /cadastro.jsp	
[04/Dec/2012:15:10:33:30] /erro.jsp	
[04/Dec/2012:15:10:33:50] /cadastro.jsp	
[04/Dec/2012:15:10:36:30] /sucesso.html	

Quadro 1: Trecho de um arquivo de *log* filtrado.

6 Análise de Log utilizando Lógica Fuzzy

A lógica *Fuzzy* propõe um cálculo matemático para traduzir e processar o conhecimento subjetivo de um especialista. É uma maneira de manipular o conhecimento com um determinado nível de incerteza [4]. O grau de pertinência é calculado através de uma função que retorna valores compreendidos entre 0 e 1. Isso indica a possibilidade do valor ou elemento pertencer a um determinado grupo [8].

Segundo a teoria da lógica *fuzzy* [8], um elemento pode pertencer a um conjunto com um grau de pertinência. Assim, um conjunto *fuzzy* A em um universo X é definido por uma função de pertinência $\mu_A(x): \rightarrow [0,1]$, de acordo com (1):

$$A = \{ \mu_A(x) / x \} x \in X \quad (1)$$

onde $\mu_A(x)$ indica o quanto x é compatível com o conjunto A . Um elemento pode pertencer a mais de um conjunto *fuzzy*, com diferentes graus de pertinência.

A primeira etapa de um sistema de inferência *Fuzzy* é denominada de Fuzzificação. Nessa etapa, os valores provenientes de um ou mais equipamentos ou sensores são transformados de variáveis numéricas (valores Crisp) em variáveis naturais [4,8]. Após o processo de Fuzzificação, dá-se início ao processo de Inferência, que tem como característica principal a utilização do conhecimento subjetivo do especialista feito através de uma base de dados [4]. No processo de Defuzzificação ocorre a transformação das variáveis linguísticas em variáveis naturais.

Assim, a lógica *fuzzy* pode ser utilizada como técnica de identificação automática de potenciais problemas de usabilidade. Para isso, os parâmetros foram estimados e pré-processados através da filtragem dos arquivos de *log* e do Modelo de Tarefas. Com esses valores amostrados, foram determinados os conjuntos universos de discurso para cada parâmetro.

6.1 Fuzzificação

A identificação de potenciais problemas de usabilidade de um sistema web baseia-se no princípio de que se a página proporciona boa usabilidade, o usuário não cometerá erros graves ou precisará de pouca ou nenhuma ajuda para a execução da tarefa. Assim, são utilizadas duas variáveis de entrada com base no conhecimento subjetivo do processo e na análise do Modelo de Tarefas. Portanto, os conjuntos que representam os potenciais problemas de usabilidade são compostos pelas duplas: {Ajuda, Erro}.

A variável linguística “Erro” é constituída por três funções de pertinência, com formato triangular, denominadas: Baixo (Pouco acesso à página de erro), Médio (Acessos médios à página de erro) e Alto (Grande número de acessos à página de erro).

A variável linguística “Ajuda” é também constituída por três funções de pertinência com formato triangular e denominadas: Baixo (Pouco acesso à página de ajuda), Médio (Acessos médios à página de ajuda) e Alto (Grande número de acessos à página de ajuda). A variável “Ajuda” foi usada no universo de discurso com intervalo de 0 a 10.

6.2 Inferência

A inferência executa operações sobre os conjuntos *fuzzy* com a combinação dos antecedentes das regras, a implicação e a aplicação do *modus ponens* generalizado [8].

Após alguns testes, foram estabelecidas 10 (dez) regras de inferência. Os consequentes são obtidos do valor mínimo da comparação entre os valores de pertinência dos antecedentes “Ajuda” e “Erro”. Para a composição de cada regra e a relação entre elas, utilizou-se a técnica de inferência MAX-MIN.

O procedimento de inferência foi baseado no modelo clássico de [8]. As regras utilizadas para a gerência do método são ilustradas na Tabela 2.

Tabela 2: Regras de Inferência.

Se	E	Então
Ajuda é baixa	Erro é baixo	Usabilidade é boa
Ajuda é alta	Erro é alto	Usabilidade é ruim
--	Erro é alto	Usabilidade é ruim
Ajuda é alta	--	Usabilidade é ruim
Ajuda é baixa	Erro é médio	Usabilidade é média
Ajuda é média	Erro é baixo	Usabilidade é média
Ajuda é média	Erro é médio	Usabilidade é média
Ajuda é média	Erro é alto	Usabilidade é ruim
Ajuda é baixa	Erro é alto	Usabilidade é ruim
Ajuda é alta	Erro é médio	Usabilidade é ruim

A partir da configuração das entradas foi gerada a superfície de classificação, ilustrada na Figura 5.

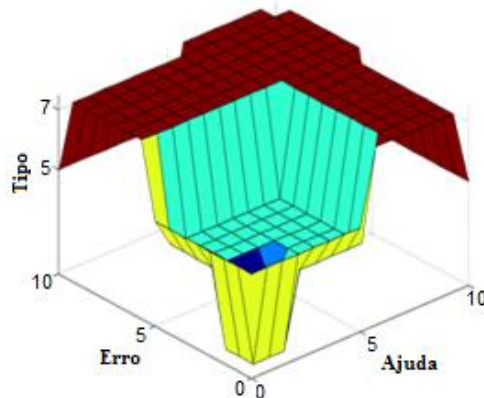


Figura 5 – Superfície de classificação gerada pelo sistema *Fuzzy*.

6.3 Defuzzificação

Nesta etapa foi determinada uma variável de classificação denominada “Problemas de Usabilidade”. As funções de pertinência para a saída da lógica *fuzzy* são mostradas na Figura 6. Nessa figura, é possível observar que a saída foi dividida em 3 classes de problemas de usabilidade: Usabilidade Boa, Usabilidade Média e Usabilidade Ruim, abrangendo um universo de discurso entre 0 e 10.

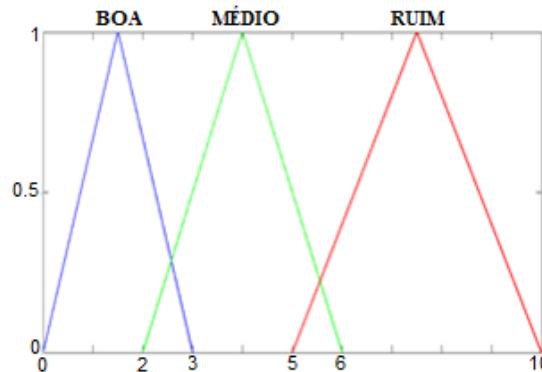


Figura 6 – Funções de pertinência da variável de saída.

Após diversos testes ficou definido que o processo de defuzzificação seria realizado através do método centroide ou bissetor de área [4,8]. A escolha desse método deu-se através de observações realizadas sobre o desempenho de classificação do sistema.

Foi analisado um arquivo de *log* e selecionadas 44 amostras que representaram a execução da tarefa definida no Modelo de Tarefas. Cada amostra indica a execução da tarefa com valores específicos de entrada, e apenas as variáveis de entrada “Erro” e “Ajuda” foram analisadas e utilizadas neste processo de classificação.

Foi obtido um resultado escalar do conjunto *fuzzy*, resultante do sistema de inferência para cada amostra. Esse valor foi analisado e de acordo com as funções de pertinência e dos intervalos definidos para cada classe de saída, foi relacionada uma das três classes de

problemas de usabilidade para cada amostra.

Com base na análise dos resultados de saída do sistema *fuzzy* e dos resultados esperados de classificação das amostras, constatou-se que a taxa de acerto de classificação foi de 96%. Assim, de 44 amostras, o sistema *fuzzy* classificou corretamente 42 amostras, sendo que as 2 amostras restantes pertenceram igualmente à duas classes diferentes de problemas de usabilidade.

7 Conclusão

Neste artigo foram apresentados os resultados obtidos com a aplicação de lógica *fuzzy* ao problema de identificação automática de potenciais problemas de usabilidade num sistema web. Taxas de acerto acima de 95% foram obtidas para todas as configurações testadas sobre o sistema *fuzzy*.

Como a usabilidade não é uma grandeza mensurável, foi necessária a construção de um modelo de tarefas para representar padrões esperados de execução da tarefa. Através desse modelo foi possível comparar os dados esperados com os dados armazenados nos arquivos de *log* do servidor. Assim, foi realizada a filtragem dos arquivos de *log* num arquivo de *log* refinado e significativo para a entrada do sistema *fuzzy*. Após diversos testes, foi observado que apenas duas variáveis (Ajuda e Erro) contribuíram significativamente para a identificação automática de problemas de usabilidade. Assim, as demais variáveis, ou seja, os acessos as demais URLs, não foram utilizadas nesse processo. Sendo assim, considerou-se apenas a quantidade de acessos às URLs: Ajuda.html e Erro.html. Essas variáveis foram suficientes para a correta classificação de problemas de usabilidade utilizando lógica *fuzzy*.

Referências

- [1] E. J. Chang, T. S. Dillon and D. Cook, An intelligent system based usability evaluation metric, IEEE Conference on Intelligent Information Systems, p. 218-226, (1997).
- [2] R. Fernandez, V. A. Correa and L. E. N. N. Patrocínio, Evaluation of estimates of effort to develop applications with supervisory system, Symposium Series In Mechatronics. p. 115-128, (2012).
- [3] D. Idoughi, M. Kerkar and C Kolski, Towards new web service based supervisory systems in complex industrial organizations: basic principles and case study, Computers in Industry, v. 61, n. 3, p. 235-249, (2010).
- [4] G. J. Klir and B. Yuan, Fuzzy sets and fuzzy logic: theory and applications. Prentice Hall, Upper Saddle River, May (1995).
- [5] M. Morandini, A. L. Schwerz and S. R. P. A Silva, Task Model Proposal for Web Sites Usability Evaluation for the Ergo Monitor Environment, Human-Computer Interaction International - Lecture Notes in Computer Science, vol. 1, p. 1188-1197, (2007).
- [6] J. Nielsen and R. L. Mack, Usability Inspection Methods, Conference companion on Human factors in computing systems, p. 413-414, (1994). DOI: 10.1145/259963.260531.
- [7] G. A. Santana, A. Goedel and L. T. E. Pansanato, Identificação Automática de Problemas de Usabilidade em Interfaces de Sistemas Web através de Redes Neurais Artificiais, Anais do XIX Congresso Brasileiro de Automática, p. 110-122, (2012).
- [8] L. A. Zadeh, Fuzzy Sets. Information and control, Elsevier, v. 8, p. 338-353, (1965).