

Simulador para treinamento de operadores do COS da CPFL baseado em inteligência computacional e tecnologias web

Rodrigo C. Araújo, RCASOFT, André L. Morelato França, UNICAMP, Fernando Antonio Caleffi, SEST ; Roberval Kaminski, Eduardo Toi – CPFL Paulista

Resumo—Este artigo descreve a concepção e metodologia de implementação de um simulador computacional para treinamento de técnicos de operação que atuam no COS da CPFL. Adotou-se uma concepção inspirada no funcionamento de sistemas especialistas. A geração de cenários pelo instrutor é realizada através da interação de duas memórias: uma memória de longo prazo e uma memória de curto prazo. A interação entre as duas memórias é propiciada por um conjunto inter-relacionado de fatos, procedimentos e regras do tipo “se A então B” que funciona como um motor de inferência que vai atualizando a memória de curto prazo com base nas decisões tomadas pelo operador, permitindo o encadeamento sucessivo de outros cenários. Na implementação utilizou-se banco de dados MySQL, plataforma Java e linguagem PHP, de tal modo que tanto a construção dos exercícios quanto a realização de sessões de treinamento seja feita através de um navegador web.

Palavras-chave—Ensino a distância, inteligência computacional, simulador para treinamento de operadores, sistemas especialistas, tecnologias web.

I. INTRODUÇÃO

A importância do desenvolvimento de novos sistemas especialistas para treinamento aumenta à medida que aumenta a complexidade das atividades e a necessidade de manter as equipes técnicas sempre prontas para novas contingências.

As contingências graves num centro de operação são raras e os técnicos operadores devem estar treinados antes mesmo que ocorram os eventos reais, daí a necessidade de se investir em sistema de treinamento que simulem a Operação do COS.

II. METODOLOGIA

O projeto consistiu no desenvolvimento de um simulador para treinamento de operadores do COS, enquadrando-se portanto na área de conhecimento chamada **operador training simulators (OTS)** na literatura mundial de sistemas de potência. Uma componente chave de qualquer simulador

para treinamento de operadores é o sistema educacional adotado, ou seja, o conjunto de métodos e ferramentas para a construção de cenários de treinamento que serão colocados à disposição do instrutor. Por essa razão, a pesquisa concentrou-se no desenvolvimento do sistema educacional do simulador, o qual foi realizado adotando-se uma metodologia inspirada no funcionamento de sistemas especialistas (técnica de inteligência artificial). A geração de cenários (situação problema apresentado ao treinando) pelo instrutor é realizada através da interação de duas memórias: uma memória de longo prazo, que contém armazenados os dados e informações mais gerais e permanentes sobre a rede elétrica e seus componentes, e uma memória de curto prazo, que vai armazenando as ações e conseqüências das ações tomadas pelo operador durante o treinamento. Ambas as memórias foram implementadas através de um banco de dados relacional MySQL. A interação entre as duas memórias é propiciada por um conjunto inter-relacionado de fatos, procedimentos e regras do tipo “se A então B” criado pelo instrutor e que funciona como um motor de inferência que vai atualizando a memória de curto prazo com base nas decisões tomadas pelo operador, permitindo o encadeamento sucessivo de outros cenários. Na metodologia adotada, um cenário é uma estrutura de dados contendo imagem, variáveis, áreas sensíveis na imagem, fatos, procedimentos e regras. As áreas sensíveis na imagem foram implementadas através de scripts em linguagem Java e JavaScript. Todos os elementos de um cenário podem ser criados e editados pelo instrutor através de telas específicas implementadas em linguagem PHP, permitindo que a interface com o usuário, seja o instrutor ao criar um exercício, seja o operador ao acessar a sessão de treinamento, possa ser acessada através de qualquer navegador de internet, facilitando enormemente o uso do simulador.

A. Cenários

O cenário contém um diagrama com a configuração do sistema elétrico em uma determinada situação que será usada como base para resolução do problema pelo operador.

B. Áreas Sensíveis

Áreas sensíveis são regiões do diagrama na qual o operador pode realizar manobras no sentido de alterar a configuração do sistema elétrico, por exemplo, disjuntores e seccionares.

Este trabalho de pesquisa faz parte do programa de P&D da CPFL, ciclo 2003/2004, sob número PD114-03.

Rodrigo C. Araújo é diretor da RCASoft (e-mail: raraujo@rcasoft.com.br).

André L. Morelato França é professor colaborador da FEEC da Unicamp (e-mail: morelato@dsee.fee.unicamp.br).

Fernando Caleffi é diretor da SEST (email:Fernando@unisoma.com).

Eduardo Toi é gerente do DOO da CPFL (email: toi@cpfl.com.br).

C. Procedimentos e Fatos

Procedimentos são as ações que o operador pode executar sobre uma área sensível (ex: abrir ou fechar um disjuntor) ou demais ações necessárias para a resolução do problema (ex: avisar o técnico do PMO). Fatos são variáveis que contêm as condições atuais do cenário e são utilizadas para acionar as regras. Os valores dos fatos são modificados pelos procedimentos.

D. Regras

Uma regra é um conjunto de condições de valores de fatos que deve ser estabelecida para seu acionamento. Por exemplo, uma regra poderia ser acionada se a carga de um transformador atingiu um determinado valor (verificando a variável – fato – correspondente) e ao mesmo tempo se um anel foi fechado. O acionamento da regra pode gerar uma mensagem para o usuário, um novo cenário ou o fim do exercício.

E. Motor de Inferência

O motor de inferência é o algoritmo usado para verificar quais as regras que devem ser executadas em um determinado momento com base nos valores atuais dos fatos (variáveis) do sistema. Caso duas ou mais regras sejam satisfeitas ao mesmo tempo, cabe ao motor de inferência decidir sobre aquela que será executada. Uma vez que uma regra é executada, o motor de inferência deve cuidar para que ela não seja executada novamente no mesmo cenário do exercício.

III. IMPLEMENTAÇÃO

Com o objetivo de tornar o sistema mais abrangente decidiu-se o focar o desenvolvimento do simulador adotando o desenvolvimento de interfaces tipo WEB compatíveis com os softwares de navegação Internet Explorer e Netscape.Navigator. Foi realizada a modelagem do fluxo de dados das aplicações e da estrutura do banco de dados. Também foi iniciada o desenvolvimento de protótipos das interfaces de alimentação do banco (criação de treinamentos) e a interface de aplicação dos treinamentos aos operadores.

Para o desenvolvimento do banco foi adotado o sistema de gerenciamento de banco de dados MySQL que é um software de domínio público. Para o desenvolvimento das interfaces gráficas e camadas de aplicação foram adotadas as linguagens de programação PHP, Java e JavaScript.

Inicialmente foi avaliada a possibilidade de utilizar a ferramenta de domínio público CLIPS como motor de inferência do sistema especialista. Mas, devido a uma limitação encontrada na ferramenta CLIPS para integração com interfaces gráficas tipo WEB, decidiu-se que, ao invés de utilizar o motor de inferência do CLIPS para processar a base de conhecimento de fatos e regras do simulador, estaríamos desenvolvendo um motor de inferência específico para o projeto, utilizando linguagem Java, JavaScript e PHP. O motor de inferência em desenvolvimento trabalha exclusivamente sobre uma base de conhecimentos composta de fatos e regras, facilitando também o processo de criação da base de conhecimento, que futuramente será realizado pelo pessoal interno da CPFL através do módulo de criação de cenários que também foi desenvolvido ao longo do projeto.

A metodologia de projeto de software adotada para o desenvolvimento do projeto foi baseada no Rational Unified Process (RUP). O RUP é uma metodologia de projeto de software desenvolvida pela Rational Software Corporation. Ela descreve como desenvolver software usando técnicas comerciais efetivamente comprovadas e define os seguintes modelos a serem seguidos pelos membros do time durante o ciclo de vida de um produto:

Desenvolvimento de software de forma interativa – considerando o tempo necessário para o desenvolvimento de sistemas sofisticados de software, não é possível definir o problema e construir a solução em um passo único. Os requerimentos geralmente mudam durante o desenvolvimento de um projeto devido a limitações da arquitetura, necessidades do usuário ou melhor entendimento do problema original. A interação permite um melhor entendimento de um projeto através de refinamentos sucessivos e coloca os itens de risco do projeto como as tarefas de maior prioridade a cada estágio da interação. Idealmente cada interação termina com um versão executável do projeto – que ajuda a reduzir os riscos do projeto, permite um maior feedback dos usuários e ajuda os desenvolvedores a se manterem focados.

Gerenciamento de requisitos – um framework de documentação é essencial para qualquer grande projeto. O RUP descreve como documentar a funcionalidade, limitações, decisões de projeto e requerimentos de negócio. Casos e cenários de uso são exemplos de artifícios descritos pelo processo que são muito eficientes tanto em capturar os requerimentos funcionais como em prover tarefas coerentes durante o desenvolvimento e implantação do sistema.

Uso de arquitetura baseada em componentes – Um arquitetura baseada em componentes cria um sistema que é facilmente extensível, promove a reutilização de software e é de entendimento intuitivo. Um componente geralmente é representado por um objeto na Programação Orientada a Objetos. O RUP provê um modo sistemático de construir este tipo de sistema, focando na produção de uma rápida arquitetura executável antes de agregar recursos de larga escala ao projeto. Os componentes são geralmente montados dentro de infra-estruturas já existentes como CORBA e COM.

Modelagem visual do software – Abstrair o programa do seu código fonte e representar o mesmo graficamente usando blocos de construção gráficos é uma forma efetiva de ter uma visão geral da solução. Isto permite que pessoas com menor conhecimento técnico tenham um melhor entendimento do problema e desta forma possam agregar um maior valor na resolução do mesmo. A UML (Unified Modeling Language) é a linguagem de representação gráfica de projetos que é usada geralmente pelo RUP.

Verificação da qualidade do software – A verificação da qualidade é um dos pontos falhos mais comuns dos projetos de software, pois é uma atividade geralmente realizada às pressas e às vezes por uma equipe diferente. O RUP auxilia no planejamento do controle e verificação da qualidade integrando estas atividades no processo e envolvendo todos os membros do time.

Controle de mudanças no software – Em qualquer pro-

reto de software as mudanças são inevitáveis e o RUP define métodos para monitorar e controlar as mudanças. Na medida que pequenas mudanças podem afetar as aplicações de forma imprevisível, este controle e monitoração das mudanças é essencial para garantir o sucesso do projeto. O RUP também define *espaços de trabalho seguros* que permitem que um programador tenha garantia que mudanças em outros sistemas não irão afetar o seu sistema. Este método é fortemente baseado nas arquiteturas de componentes.

IV. RESULTADOS

Os principais resultados obtidos pelo projeto estão enumerados abaixo:

1. Definição do Escopo e Concepção Geral do Projeto

Período inicial de concepção do escopo do projeto, onde foi definido que o foco das atividades seria o desenvolvimento de um sistema especialista utilizando conceitos de Inteligência Artificial. Concebia-se um sistema que simulasse o sistema supervisorio e acabou-se construindo um sistema de aprendizagem que simula as atividades do operador.

Como o sistema possui dois módulos, o módulo do operador e o módulo do instrutor, dando a CPFL a possibilidade de elaborar os exercícios que forem necessários, criar a complexidade desejada e mantê-los sempre atualizados.

2. Estudo das Tecnologias de Inteligência Artificial

Aprofundamento nas tecnologias de Inteligência Artificial que seriam utilizadas para o desenvolvimento, particularmente o CLIPS, e como estas tecnologias se interagiriam com outras linguagens de programação orientadas a objetos, como C++ e Java, que seriam utilizadas para a implementação do front-end do sistema (interfaces gráficas com o usuário).

3. Modelagem de uma Modalidade Inicial de Treinamento

Nesta fase foi feita a implementação de um primeiro modelo de treinamento dos operadores de COS. Para isso foi escolhido um dos documentos de descrição dos procedimentos operativos que foram fornecidos pela CPFL e utilizando conceitos de Inteligência Artificial foi elaborado o modelo mais adequado de aplicação de treinamento aos operadores para aprendizado dos procedimentos operativos contidos neste documento. Após a definição do modelo, foi feita a implementação de sua base de conhecimento e fluxo funcional utilizando em um primeiro momento a interface tipo texto do CLIPS. Após a implementação, o modelo foi apresentado para a CPFL para refinamento da base de conhecimento e realização de ajustes.

4. Análise da Documentação Existente e Conhecimentos não Documentados para Levantamento das Metodologias de Treinamento

Nesta fase foi feita uma análise detalhada de toda a documentação existente na CPFL que é utilizada na instrução dos operadores do COS e também de conhecimentos tácitos necessários que não estão documentados, mas que são importantes e deverão ser contemplados pelo sistema especialista. A partir desta análise foram elaboradas as metodologias de treinamento que serão implementadas no sistema especialista. Como diversos tipos de conhecimento deverão ser transmitidos para os operadores, foram escolhidas metodologias de treinamento adequadas para cada tipo de conhecimento. Por exemplo, um determinado tipo de conhecimento poderá ser transmitido em um treinamento tipo pergunta e resposta, enquanto que outro tipo de conhecimento seria mais bem trabalhado em um treinamento que gerasse um cenário para simulação do comportamento do sistema elétrico.

5. Modelagem dos Objetos/Classes da Base de Conhecimento

A partir da definição das metodologias de treinamento que seriam aplicadas pelo sistema especialista, foi feita a modelagem dos objetos e classes que constituiriam a base de conhecimento. O objetivo desta modelagem foi criar um padrão para estruturação do conhecimento do sistema especialista, de modo que ele possa utilizar seu motor de inferência baseado em fatos e regras para processar este conhecimento.

durante a aplicação dos treinamentos aos operadores do COS.

6. Desenvolvimento da Interface de Alimentação da Base de Conhecimento

Uma vez definido o modelo de objetos e classes da base de conhecimento, foi implementada uma interface gráfica para possibilitar a alimentação das informações. Desta forma, a base de conhecimento do sistema especialista pode ser facilmente revisada para refinamento das informações existentes e atualização de novas informações. A própria CPFL pode fazer a alimentação das informações da base de conhecimento.

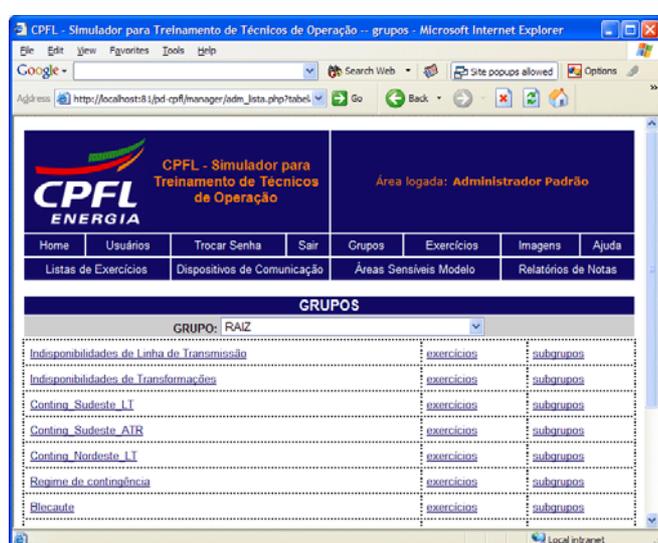


Figura 01 – Interface de Alimentação da Base de Conhecimento

7. Carga e Refinamento das Informações da Base de Conhecimento

Nesta fase foi feita a alimentação das informações da base de conhecimento do sistema especialista. Esta foi uma das partes mais importantes do projeto, pois a qualidade e precisão das informações fornecidas é que determina o grau de inteligência e eficácia dos treinamentos do sistema especialista. Esta fase foi feita em interação constante com a CPFL.

8. Desenvolvimento do Módulo de Aplicação dos Treinamentos

Esta fase foi feita em paralelo a alimentação das informações da base de conhecimento do sistema especialista. Nela foram implementadas as rotinas de programação do sistema necessárias para aplicação de cada metodologia de treinamento definida na quarta etapa. Além das rotinas de programação do fluxo funcional do sistema especialista, foram desenvolvidas interfaces gráficas com o usuário para tornar o sistema mais atraente, amigável e interativo. Estas interfaces gráficas foram desenvolvidas em linguagem PHP, Java e

JavaScript, possibilitando a concepção de interfaces tipo WEB, tornando o sistema acessível de forma remota, sem a necessidade de instalação local de arquivos no computador do usuário.

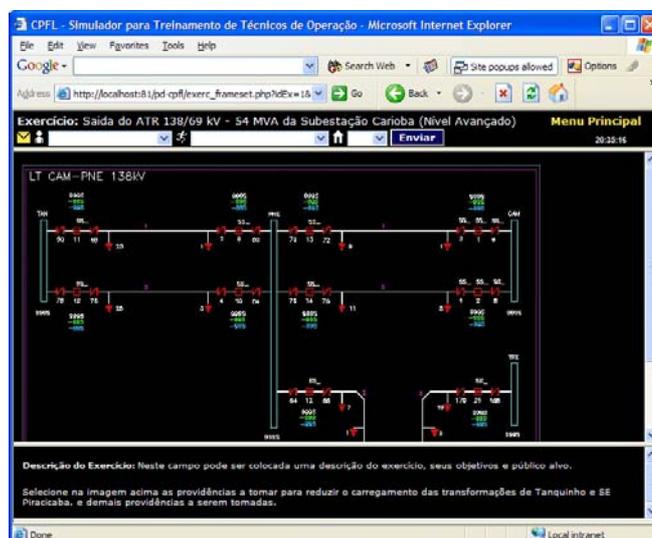


Figura 02 – Módulo de Aplicação dos Treinamentos

9. Testes do Sistema Junto ao Usuário Final

Nesta fase o sistema foi disponibilizado para utilização pelos operadores do centro de operação com o objetivo de se realizar ajustes que fossem necessários na base de conhecimento, levantamento e correção de bugs, e validação da eficácia dos treinamentos aplicados pelo sistema especialista.

10. Edição dos exercícios

Para facilitar a tarefa do instrutor de criar os exercícios, é oferecida a função de copiar, colar e alterar exercícios, cenário e funções. Quando se copia um cenário são transportadas todas as imagens, áreas sensíveis, fatos, procedimento e regras.

11. “Log”

Todas as ações dos operadores ficam registradas num “log”, oferecendo ao instrutor as informações necessárias para acompanhar o treinamento e avalia-los.

12. Implantação Final do Sistema e Preparação da Documentação e Manuais do Usuário

Esta última fase compreendeu a implantação do sistema para utilização por todos os operadores do centro de operação da CPFL. Foram preparados os documentos técnicos e manuais relacionados a instalação, distribuição, configuração, carga de informações e operação de todos os módulos do sistema. O manual de operação também faz parte do sistema, e está em WEB.

V. CONCLUSÕES

Foi desenvolvida uma ferramenta para treinamento muito mais que um sistema simulador de operadores de COS, uma vez que ele pode ser utilizado a treinamento de outras atividades.

VI. AGRADECIMENTOS

Agradecemos a participação do técnico João Paulo Parreira, do operador de sistema Paulo César Gonçalves e da Engenheira Cleide Paganin Baisi, por acompanhar nossas reuniões, pelas freqüentes contribuições técnicas, pelas sugestões praticas e pela criação de inúmeros exercícios que foram implementados.

VII. REFERÊNCIAS BIBLIOGRÁFICAS

Periódicos:

- [1] J. F. Fuller, E. F. Fuchs, and K. J. Roesler, "Influence of harmonics on power distribution system protection," *IEEE Trans. Power Delivery*, vol. 3, pp. 549-557, Apr. 1988.
- [2] E. H. Miller, "A note on reflector arrays," *IEEE Trans. Antennas Propagat.*, a ser publicado.
- [3] R. J. Vidmar. (1992, Agosto). On the use of atmospheric plasmas as electromagnetic reflectors. *IEEE Trans. Plasma Sci.* [Online]. 21(3), pp. 876-880. Disponível: <http://www.halcyon.com/pub/journals/21ps03-vidmar>

Livros:

- [4] E. Clarke, *Circuit Analysis of AC Power Systems*, vol. I. New York: Wiley, 1950, p. 81.
- [5] G. O. Young, "Synthetic structure of industrial plastics," in *Plastics*, 2nd ed., vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15-64.
- [6] J. Jones. (1991, May 10). *Networks*. (2nd ed.) [Online]. Disponível: <http://www.atm.com>

Relatórios Técnicos:

- [7] E. E. Reber, R. L. Mitchell, and C. J. Carter, "Oxygen absorption in the Earth's atmosphere," Aerospace Corp., Los Angeles, CA, Relatório Técnico. TR-0200 (4230-46)-3, Nov. 1968.

Artigos Apresentados em Conferências (Não publicados):

- [8] D. Ebehard and E. Voges, "Digital single sideband detection for interferometric sensors," apresentado na 2nd Int. Conf. Optical Fiber Sensors, Stuttgart, Alemanha, 1984.

Artigos em Anais de Conferências (Publicados):

- [9] J. L. Alquieres and J. C. Praca, "The Brazilian power system and the challenge of the Amazon Transmission," in *Proc. 1991 IEEE Power Engineering Society Transmission and Distribution Conf.*, pp. 315-320.

Dissertações e Teses:

- [10] S. Hwang, "Frequency domain system identification of helicopter rotor dynamics incorporating models with time periodic coefficients," Tese de Doutorado, Dept. Aerosp. Eng., Univ. Maryland, College Park, 1997.

Normas:

- [11] *IEEE Guide for Application of Power Apparatus Bushings*, IEEE Standard C57.19.100-1995, Aug. 1995.

Patentes:

- [12] G. Brandli e M. Dick, "Alternating current fed power supply," U.S. Patent 4 084 217, 4 de novembro, 1978.