



**XX SNPTEE
SEMINÁRIO NACIONAL
DE PRODUÇÃO E
TRANSMISSÃO DE
ENERGIA ELÉTRICA**

Versão 1.0
XXX.YY
22 a 25 Novembro de 2009
Recife - PE

GRUPO IX

GRUPO DE ESTUDO DE OPERAÇÃO DE SISTEMAS ELÉTRICOS - GOP

**SIMULADOR DE AUTOMATISMOS E PROCESSOS
PARA TREINAMENTO DE OPERADORES DE USINAS HIDRELÉTRICAS**

**Samuel Domingos Maganeti Lazzarin(*)
ALSTOM**

RESUMO

Para garantir a qualidade da supervisão e controle de uma usina hidrelétrica, é vital que seus operadores conheçam plenamente, tanto o funcionamento dos processos dos diversos subsistemas da planta, quanto os automatismos do Sistema Digital de Supervisão e Controle inerentes a cada subsistema. Para garantir a competência destes operadores, simuladores são utilizados para treinamento de novos profissionais. Este artigo pretende explorar um conceito diferenciado desenvolvido pela Alstom para simuladores de treinamento de operadores de usinas hidrelétricas. Neste novo conceito, juntamente aos modelos de processo dos equipamentos, são simulados também os próprios automatismos do Sistema Digital de Supervisão e Controle.

PALAVRAS-CHAVE

Simulação, tradutor, Sistema Digital de Supervisão e Controle

1.0 - INTRODUÇÃO

O Sistema Digital de Supervisão e Controle (SDSC) de usinas hidrelétricas possui interface com inúmeros equipamentos distribuídos pela planta. Enquanto a supervisão é feita por sensores e medidores, os comandos provenientes do SDSC originam-se de dois grandes grupos: pedidos de comando feito pelo do operador através de Interfaces Homem-Máquina (IHM), assim como através de algoritmos automatizados que tomam iniciativas pré-determinadas dado certo cenário de operação da usina. Independente do grupo em que se encaixam os comandos são sempre intertravados e liberados para o campo após a verificação das condições de seu acionamento.

Olhando por este prisma, o SDSC é uma ferramenta utilizada pelo operador da usina para avaliar de maneira conjunta o processo de geração de hidroeletricidade. As manobras, automatizadas nos controladores lógicos programáveis ou comandadas pelo próprio operador, são fruto de uma interpretação contínua do estado de cada subsistema composto pelos diversos equipamentos da planta.

Portanto, para formação de um bom operador de usina hidrelétrica é, insuficiente que o mesmo conheça apenas a operação individual e local de cada equipamento (i.e. bombas, chaves ou reguladores). Entender o automatismo atrelado ao processo como um todo, é extremamente necessário para uma saudável operação da usina.

A Alstom, em sua mais nova experiência com simuladores de treinamento, inovou na concepção destes simuladores de processo, adicionando à modelagem do simulador, uma camada a mais que simula os próprios algoritmos executados nos controladores lógicos programáveis. Esta inovação, detalhada ao longo deste informe, permitiu que fosse criada uma nova metodologia conceitual para o desenvolvimento de simuladores de processo, agregou valor e confiabilidade ao produto.

2.0 - ARQUITETURA DO CENTRO DE TREINAMENTO

A arquitetura de computadores, redes e software é exibida na Figura 1.

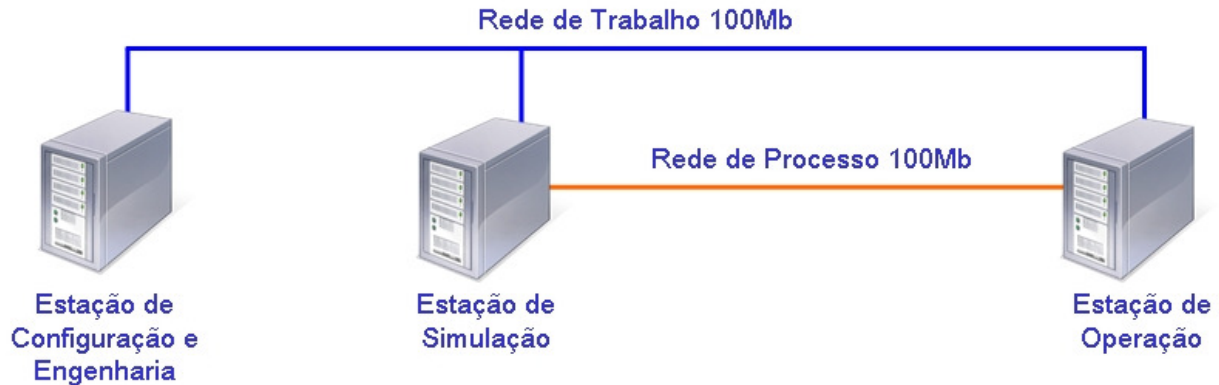


Figura 1 – Arquitetura do centro de treinamento

A arquitetura é composta por três estações de trabalho:

- A Estação de Configuração e Engenharia: é através desta estação que o projeto do Sistema Digital de Controle e Supervisão é implementado. De maneira centralizada esta estação concentra toda a configuração da rede e dos componentes que fazem parte da arquitetura de supervisão e controle da usina, como por exemplo: telas de interface homem-máquina, algoritmos de controladores e configuração da base de dados de tempo real.
- A Estação de Operação: é através dela que a planta é supervisionada e controlada pelo operador. Possui o banco de dados de tempo real e interface homem-máquina.
- Estação de Simulação: que concentra a simulação de todo o processo de geração de hidroeletricidade através de modelos de equipamentos assim como

Estas três estações são interligadas por duas redes:

- A Rede de Trabalho: rede TCP/IP através da qual são distribuídas as configurações da Estação de Configuração e Engenharia para a Estação de Simulação e a Estação de Operação.
- A Rede de Processo: rede TCP/IP dedicada ao tráfego dos dados entre a Estação de Simulação e a Estação de Supervisão e Controle através do protocolo IEC104.

Na Estação de Configuração e Engenharia assim como na Estação de Operação estão instalados os pacote do ALSPA P320® Plant Automation – solução Alstom para Sistemas Digitais de Supervisão e Controle. Já na Estação de Simulação está instalado o software RSI SIMCON - INDISS™ Simulation Plataform, o qual oferece um ambiente gráfico para instanciar os modelos de simulação produzidos em linguagem C++, assim como determinar interfaces existentes entre modelos.

3.0 - AS CAMADAS DO SIMULADOR

Com o intuito de reduzir o tempo de programação e evitar retrabalho durante a implementação e configuração do Sistema de Treinamento de Operadores, concebeu-se uma estrutura de simulação dividida em camadas independentes.

O conceito de camadas resulta em uma classificação criteriosa e adequada dos vários modelos de simulação desenvolvidos para serem instanciados na Estação de Simulação.

A Figura 2 detalha o posicionamento e função de cada camada. Começando pelo nível 0, tem-se na base inferior o processo físico. Nesta camada são simulados os fenômenos hidráulicos, mecânicos ou elétricos de primeira

ordem dos equipamentos eletromecânicos que compõe a usina, por exemplo: disjuntores, chaves, bombas, reservatórios, etc.

Acima do processo físico existem os sinais de entrada e saída, isto é, nesta camada são modeladas as interfaces que os modelos da camada inferior possuem com o Sistema Digital de Supervisão e Controle. Por exemplo: os ventiladores da usina possuem indicações de ligado, presença de fluxo de ar, disjuntor térmico disparado, etc.

Em seguida está o nível 1 que também é simulado. Sua primeira camada é chamada de Cartas de Interface. Sua função é modelar o mapeamento dos bornes de interface de cada módulo de entradas e saídas do controlador lógico programável. Por exemplo, um módulo de 32 entradas digitais é modelado com 32 variáveis do tipo booleana, já um módulo de 16 entradas analógicas, é modelado com 16 variáveis do tipo float.

Acima se encontra a camada que modela os algoritmos de automação. Esta camada é responsável por simular a lógica criada para os controladores lógico programáveis que automatizam a planta em seus diversos subsistemas. É esta a camada que recebe enfoque considerável neste trabalho e que será contextualizada nos próximos itens.

No sentido ascendente, encontra-se a camada que selecionará, a partir do modelo de algoritmos, as variáveis para o banco de dados de tempo real. Este subconjunto de variáveis são então transmitidas via gateway IEC104 para a Estação de Operação onde o banco de dados de tempo real está sendo executado.

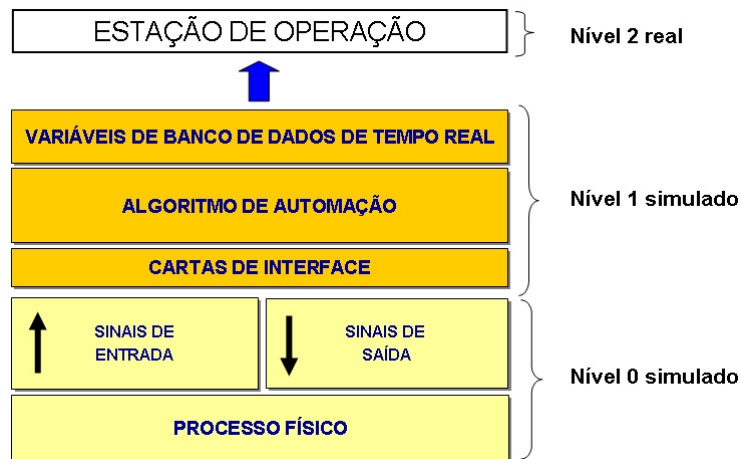


Figura 2 - Camadas de simulação

4.0 - CRIAÇÃO DOS MODELOS

As camadas anteriormente descritas são compostas por modelos elaborados em linguagem C++. Todos os modelos disponíveis para simulação seguem uma estrutura de criação. Obedecendo esta estrutura, os modelos ficam aptos a comunicar-se entre si, seja entre modelos de uma mesma camada, assim como entre modelos de camadas vizinhas.

A Tabela 1 expõe a estrutura para os modelos de simulação.

Tabela 1 - Estrutura do modelo

Preâmbulo
Declaração de variáveis em linguagem C++ como: int, float, boolean, char, dentre outras disponíveis nesta linguagem de programação. Estas variáveis serão utilizadas internamente nas lógicas booleanas, assim como com os operadores matemáticos.

Mapeamento
<p>Dentre todas as variáveis declaradas, são definidos 3 grandes grupos dentro deste mapeamento:</p> <ul style="list-style-type: none"> • Variáveis de entrada – são aquelas que receberão valores vindos de outras instâncias de modelos; • Variáveis de saída – são aquelas variáveis calculadas dentro do modelo e que servem de saída para “alimentar” outras instâncias de modelos; • Variáveis internas – são variáveis auxiliares utilizadas internamente no modelo e não ficam disponíveis para uso de outras instâncias de modelo.
Inicialização do modelo
<p>Este algoritmo é executado somente no primeiro ciclo de simulação. No primeiro ciclo os valores iniciais das variáveis declaradas no preâmbulo são repassados ao modelo.</p>
Rotina de simulação
<p>Diferentemente da inicialização do modelo, este trecho de algoritmo é executado recursivamente a cada ciclo de simulação. Aqui são executadas as lógicas que simulam o comportamento de cada modelo frente às condições impostas pelas variáveis de entrada de cada modelo. É aqui também que as variáveis de saída são calculadas e disponibilizadas para as outras instâncias de modelos.</p>
Finalização da simulação
<p>Este algoritmo é executado somente no último ciclo de simulação, quando o simulador é parado. No último ciclo é possível atribuir valores finais às variáveis definidas no preâmbulo de modo que a simulação sempre finalize em uma condição específica.</p>

4.1 Exemplo de um modelo

Para exemplificar a estrutura de criação de um modelo, na Tabela 2 é exposto em linhas gerais o modelo de um disjuntor telecomandável.

Tabela 2 - Exemplo de modelo de um disjuntor telecomandável

Preâmbulo
<pre>int DJ_fechado; int DJ_aberto; int DJ_falha; int DJ_comando_abrir; int DJ_comando_fechar ; int i, j; float DJ_tensao_montante; float DJ_tensao_jusante;</pre>
Mapeamento
<p>Variáveis de entrada: DJ_comando_abrir, DJ_comando_fechar, DJ_montante; Variáveis de saída: DJ_aberto, DJ_fechado, DJ_falha, DJ_jusante; Variáveis internas: i, j.</p>
Inicialização do modelo

DJ_aberto = 1; DJ_fechado = 0; DJ_falha = 0; i = 0; j = 0; DJ_tensao_jusante = 0.0; DJ_tensao_montante = 0.0;
Rotina de simulação
Se (DJ_comando_abrir = 1) OU (DJ_falha = 1) então DJ_aberto = 1; DJ_fechado = 0; Fim Se Se (DJ_comando_fechar = 1) E (DJ_falha = 0) então DJ_aberto = 0; DJ_fechado = 1; Fim Se Se (DJ_fechado = 1) E (DJ_aberto = 0) então DJ_tensao_jusante = DJ_tensao_montante; Fim Se Se (DJ_fechado = 0) E (DJ_aberto = 1) então DJ_tensao_jusante = 0.0; Fim Se
Finalização da simulação
DJ_aberto = 1; DJ_fechado = 0; DJ_falha = 0; i = 0; j = 0;

Portanto, são variáveis de saída do modelo aquelas que descrevem o estado do disjuntor (aberto, fechado, e presença de falha), por outro lado são variáveis de entrada os comandos abrir ou fechar, que segundo o algoritmo criado, são intertravados com o estado de falha do equipamento.

Para finalizarmos a exemplificação, suponhamos que este disjuntor alimente um barramento de corrente contínua a partir de um banco de baterias. O modelo das baterias faria interface com o modelo deste disjuntor transferindo a tensão das baterias à variável DJ_tensao_montante. Ao mesmo tempo, o disjuntor faria interface com o barramento, transferindo a variável DJ_tensao_jusante para uma variável de entrada do modelo do barramento.

5.0 - O MODELO DE AUTOMATISMO

O modelo de automatismo segue a mesma estrutura mostrada e exemplificada anteriormente. Sua inserção se dá na medida em que se torna interessante simular o comportamento dos controladores que supervisionam e enviam comandos para a usina, sejam estes, a pedido do operador ou automatizados dada certa condição de operação da usina.

A camada onde os modelos de automatismo estão inseridos permite que estes tenham interface com: os modelos de carta de interface, e com o modelo de variáveis mapeadas para a base de dados de tempo real e também com os outros modelos de automatismos dentre os diversos que compõe a usina.

O modelo de cartas de interface faz o papel dos módulos do CLP de entrada e saída que monitoram recebendo e enviando sinais elétricos (digitais e analógicos) aos equipamentos de que fazem parte da usina. Estes sinais elétricos são mapeados em variáveis ordenadas conforme afetação dos pontos de aquisição e controle.

Portanto, o modelo das cartas de interface somente ordena os dados trocados com os equipamentos de campo. Internamente, o modelo de automatismo irá executar o mesmo algoritmo executado nos controladores lógicos

programáveis reais da usina. Neste modelo, serão geradas diversas variáveis internas que originarão informações consolidadas para serem repassadas à camada de base de dados de tempo real, assim como serão calculados os comandos que devem ou não ser enviados para os modelos de equipamento do campo através do modelo de cartas de interface.

O modelo de automatismo ainda possui como variáveis de entradas as informações que provêm da base de dados de tempo real, que são os pedidos de telecomando ou setpoints feitos através da estação de operação. O modelo de base de dados de tempo real captura os pedidos de telecomandos que transitam via gateway IEC104 e os mapeiam de maneira organizada para serem usados na camada dos modelos de automatismo.

Nota-se, portanto, que a camada dos modelos de automatismo possui as mesmas interfaces que um controlador real teria em sua aplicação dentro de uma hidrelétrica.

6.0 - O PROCESSO DE CRIAÇÃO DO MODELO DE AUTOMATISMO

Para que o conceito de simulação fosse obedecido, o algoritmo presente na camada de simulação de automatismos deve ser o mesmo implementado nos controladores reais de primeiro nível do Sistema Digital de Supervisão e Controle. Esta necessidade fomentou a criação de um processo automatizado que traduzisse o código implementado nos controladores para a linguagem C++ utilizada nos algoritmos do simulador.

Desenvolvido em linguagem C++, este tradutor é utilizado conforme a Figura 3:

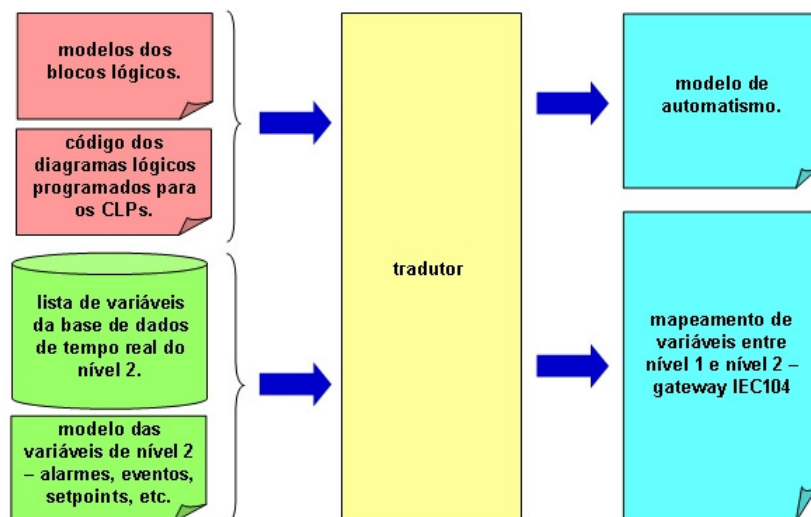


Figura 3 - Utilização do tradutor

A solução Alstom para Sistemas Digitais de Controle e Supervisão possui um software de engenharia que concentra na Estação de Configuração e Engenharia, toda as configurações das estações de trabalho de nível 2 e também de todos os controladores de nível 1.

A partir deste software de engenharia é possível exportar-se em formato texto as diversas configurações implementadas no projeto do Sistema Digital. Para utilizar o tradutor de linguagens, são exportados em formato texto os códigos dos diagramas lógicos programados para os controladores de nível 1, assim como a lista de variáveis de base de dados de tempo real que serão utilizadas nas IHMs de nível 2.

Para finalizar o conjunto de informações necessárias utilizadas como entrada para o tradutor de linguagens somam-se a estes dois arquivos exportados: o modelo em C++ dos blocos lógicos (E, OU, SET-RESET, XOR, NOT, etc.) utilizados na programação dos controladores lógicos, assim como a lista dos tipos de variáveis de nível 2 que constituem o banco de dados de tempo real para as IHMs (alarmes, eventos, setpoints, telecomandos, etc). Estas duas últimas informações adicionadas são criadas uma única vez durante a concepção do tradutor de linguagens.

Utilizando estas informações, o tradutor é capaz de gerar dois outros arquivos textos: o modelo de automatismo, isto é, o código em C++ equivalente a lógica implementada nos controladores da usina e a lista de variáveis que

devem ser transmitidas via gateway IEC104 e, por fim, alimentar o banco de dados de tempo real das IHMs do operador que será treinado.

O código traduzido é então incluído na rotina de simulação do modelo de automatismo conforme indicado na Tabela 1, enquanto a lista de variáveis transmitidas via IEC104 é incluída no preâmbulo e mapeamento do modelo de variáveis de banco de dados de tempo real.

7.0 - CONCLUSÕES

O uso do tradutor de linguagens agrega fidelidade no desenvolvimento do simulador de treinamento na medida em que se reduz a zero a possibilidade de haver diferenças do automatismo implementado nos controladores em reais e nos modelos de automatismos que constituem uma das camadas do simulador. Além disso, o uso do tradutor é conveniente, pois as freqüentes modificações no projeto real do Sistema Digital são facilmente transportadas para o simulador de treinamento, portanto, não há necessidade de se atualizar dois algoritmos separadamente.

A simulação de todos os subsistemas pertencentes a usina permite que o processo de hidroeletricidade seja melhor concebido pelo operador que está recebendo o treinamento, isto é, unidades geradoras, serviços auxiliares elétricos e mecânicos e subestações associadas são simulados de maneira conjunta e coerente.

Como efeito colateral, mas não menos importante, notou-se que o uso do tradutor de linguagens criava um feedback consistido do automatismo criado para o projeto real. Diversas foram as situações que o automatismo real precisou ser alterado frente a sua implementação incorreta. Incoerências eram notadas nos testes de simulação conjunta, onde os modelos de automatismos eram simulados junto com o conjunto de equipamentos da planta.

8.0 - REFERÊNCIAS BIBLIOGRÁFICAS

(1) ALSPA P320® Plant Automation, Março 2009

<http://www.power.alstom.com/home/equipment___systems/control_systems_products___services/plant_automation>

(2) RSI SIMCON - INDISS™ Simulation Platform, Março 2009

<http://www.simulationrsi.net/RSI-products_services-indiss.htm>

9.0 - DADOS BIOGRÁFICOS

Samuel D. M. Lazarin, natural de Piracicaba, nascido em 1982, graduado como Engenheiro Eletricista de Energia e Automação Elétricas pela Escola de Engenharia de São Paulo, POLI-USP.