

Projeto P&D D-241 ANEEL (CEMIG/Concert/UFMG)
Desenvolvimento de Software para Conversão Automática de Diagramas de
Operação de Subestações em Quadros Sinóticos e Base de Dados Digitalizada

Tema do relatório: Etapa 12c - Manual de utilização e
operação do sistema

Coordenação Técnica:

Renato Cardoso Mesquita (UFMG)
Bernadete Maria Mendonça Neta (CEMIG)
Daniel Ramos (Concert)

Departamento de Engenharia Elétrica
Universidade Federal de Minas Gerais
31 de Maio de 2011

Equipe (UFMG)

Professor:

Renato Cardoso Mesquita (Coordenação técnica);

Alunos:

Bruno Guedes Azevedo Viana (Graduação em Ciência da Computação)
Naísses Zoia Lima (Doutorado em Engenharia Elétrica)

Resumo

Este relatório se refere à etapa 12(c) do ano 3 do projeto P&D D-241 (CEMIG/Concert/UFMG), *“Desenvolvimento de Software para Conversão Automática de Diagramas de Operação de Subestações em Quadros Sinóticos e Base de Dados Digitalizada”*. Nesta etapa estava prevista a entrega da documentação de utilização e operação do sistema desenvolvido pela UFMG.

Sumário

1. Introdução	5
2. Instruções de uso do programa	6
3. Node Templates	8
4. Conclusões	12
5. Referências Bibliográficas	13

1. Introdução

Conforme foi descrito em [MES08], o objetivo desse projeto de P&D é desenvolver uma ferramenta para Geração Automática de Quadros Sinóticos e Base de Dados dos Sistemas SCADA e EMS (Energy Management System) a partir dos arquivos de Descrição de Configuração das Subestações da CEMIG.

Isto é feito seguindo uma série de etapas bem definidas, que englobam diversos softwares distintos para realização de tarefas que se completam para cumprir o objetivo final. Estas etapas são:

1-Inicialmente o usuário deve esboçar a subestação utilizando o Configurador de Sistemas, desenvolvido pela Concert Technologies, obedecendo a certas regras de desenho.

2-Então, o Configurador de Sistemas se comunica com o Gerador de Sinóticos, desenvolvido pela UFMG, através de um arquivo XML. Os dados são transferidos de um programa ao outro, de modo que o Gerador de Sinóticos processe as informações inseridas pelo usuário, e crie um arquivo descritor da tela referente à subestação – este arquivo também é no formato XML.

3-Por fim, o arquivo XML criado pelo Gerador de Sinóticos pode ser processado por um terceiro programa, que fica encarregado de produzir o desenho da tela em si. Este programa posiciona todos os elementos gráficos nas posições definidas pelo XML, utilizando os padrões utilizados internamente pela CEMIG. Este programa está sendo desenvolvido pela empresa Bentley, na forma de uma macro para o programa Microstation.

O Gerador de Sinóticos é o programa que foi desenvolvido pela UFMG e é ele que estamos documentando neste manual. Ele obedece a regras específicas de comunicação com os programas que o envolvem e são essas regras que estão aqui documentadas. Não há nenhuma comunicação do programa com o usuário. A interação com o usuário somente se faz nos demais módulos, ou seja, não há qualquer operação do usuário aqui detalhada.

2. Instruções de uso do programa

O programa desenvolvido é fornecido na forma de um executável para o ambiente Windows de nome **Gerador.exe**.

Na sua execução ele lê um arquivo com a configuração de uma subestação gerado pelo programa “Configurador de Sistemas” desenvolvido pela Concert Technologies, em formato SCD, um arquivo XML com a configuração dos Node Templates e um arquivo XML com a configuração dos elementos conhecidos (transformadores, chaves, disjuntores, barramentos, etc).

Ele gera duas saídas: um arquivo XML, que descreve o desenho da subestação e será usado como entrada para o macro QuadroSinoticos do MicroStation que está sendo desenvolvido pela empresa Bentley; e um arquivo GML que serve para verificação interna, podendo ser desenhado em programas visualizadores de arquivos GML (Graph Modeling Language) como, por exemplo, o Yed Graph Editor.

O formato desses arquivos já foi detalhado em relatórios anteriores.

Para executar o programa a sintaxe é:

Gerador.exe [Entrada SCD] [XML de NodeTemplates] [XML de elementos] [nome da saída] [XML de vãos]

* [Entrada SCD] é o caminho para o arquivo SCD contendo a descrição da subestação.

* [XML de NodeTemplates] é o caminho para o arquivo XML contendo a estrutura dos NodeTemplates mapeados.

* [XML de elementos] é o caminho para o arquivo XML contendo a estrutura dos elementos conhecidos.

* [nome da saída] é o caminho para os arquivos .gml e .xml criados. Ambos serão criados com o mesmo nome, mas extensões diferentes.

* [XML de vãos] é o caminho para o arquivo XML contendo a estrutura dos vãos mapeados (não necessariamente implementado, mas incluso no código).

Exemplo de uso:

Gerador.exe In\BHBN.SCD Conf\NodeTemplates.xml Conf\elementos.xml Out\saida-BHBN Conf\VaosTemplates.xml

Neste caso,

- BHBN.SCD é o arquivo de entrada contendo a descrição da subestação (no caso a subestação BHBN da CEMIG). Este arquivo está no subdiretório In do diretório onde o executável Gerador.exe está localizado.
- NodeTemplates.xml é o arquivo com a estrutura dos node templates mapeados. Este arquivo está no subdiretório Conf do diretório onde o executável Gerador.exe está localizado.

- elementos.xml é o arquivo com a estrutura dos elementos conhecidos. Este arquivo está no subdiretório Conf do diretório onde o executável Gerador.exe está localizado.
- saída-BHBN é nome dos arquivos .gml e .xml criados. Ambos serão criados com o mesmo nome, mas extensões diferentes. Estes arquivos serão criados no subdiretório Out do diretório onde o executável Gerador.exe está localizado.
- VaosTemplates.xml é o nome do arquivo onde os vãos estão mapeados. Também localizado no subdiretório Conf do diretório onde o executável Gerador.exe está localizado.

Para mapeamento dos NodeTemplates tem-se a sintaxe descrita no item 3, a seguir.

3. Node Templates

Um Node Template corresponde a um conjunto específico de elementos (equipamentos e barramentos) interligados de uma maneira particular. É descrito por meio de um código XML em um arquivo que contém os mapeamentos de todos os Node Templates existentes em todas as telas. O procedimento de mapeamento de Node Templates acompanhado de um exemplo (Figura 1) é descrito mais à frente nesta seção.

Node Templates foram criados para padronizar configurações pré-definidas e normatizadas de elementos interconectados e que se repetem em várias instâncias de tela. Dessa forma, garante-se que essas configurações serão sempre desenhadas da mesma forma, de maneira correta e consistente, assim como atenderão os padrões de desenho de telas especificados em [MEN08a].

Ao ler as informações de um arquivo SCD gerado pelo Configurador de Sistemas, o Gerador busca os Node Templates (disponíveis no arquivo XML de Node Templates) correspondentes às topologias de ligação entre elementos da subestação. Durante esse processo, se um Node Template não é encontrado no arquivo XML, isto é, se não existe um mapeamento para uma das topologias de elementos da subestação fornecidas no arquivo SCD, então faz-se necessário criar o mapeamento do Node Template correspondente à topologia. Após o processo de identificação pelo Gerador dos Node Templates de uma subestação, estes são conectados aos seus respectivos barramentos. Por fim, aplica-se o algoritmo de Sugiyama para posicionar todos os elementos na tela.

Como afirmado anteriormente, o mapeamento de Node Templates é feito em um arquivo XML. Cada Node Template é mapeado através da tag <Node> e traz as seguintes informações:

- <Equipment /> : Equipamento interno ao nó.
- <Bus /> : Barramento interno ao nó.
- <Edges /> : Arestas que interligam elementos internos do nó.
- <ExternalPoint />: Pontos de ligação externa. Constituem os pontos de ligação entre o nó e elementos externos (que não fazem parte do nó). Pontos de ligação externa também podem ser parte de um barramento.
- <ExternalArea />: Áreas de ligação externa. Consiste em uma área onde pontos de ligação podem ser criados dinamicamente (usada em *templates* que fazem ligação externa por meio de barramentos, que podem receber um número imprevisível de ligações externas).

Cada nó possui um identificador (*id*) único composto pela soma dos elementos que contém, além do comprimento (*w*) e altura (*h*).

Cada equipamento possui um número inteiro que representa um identificador (*id*) interno ao nó, uma descrição (*desc*) de qual equipamento se trata, seu tipo (*type*), comprimento (*w*) e altura (*h*) (conforme especificado em [MEN08]), suas posições x (*x*) e y (*y*) internas (ligadas ao canto superior esquerdo da célula do equipamento), sendo a referência do sistema de coordenadas tomada como o canto superior esquerdo da célula correspondente ao nó. Além disso, tem-se também o ângulo do elemento (*ang*), definido como um valor múltiplo de 90 graus (0, 90, 180 ou 270), que indica o ângulo de rotação no sentido anti-horário que o elemento irá sofrer. Deve ser notado que caso haja rotação do equipamento, seu ponto de origem também é rotacionado de acordo com o ângulo de rotação, isto é, se o equipamento sofre uma rotação de 90 graus, então o ponto de origem deixa de ser o ponto superior esquerdo e passa a ser o inferior esquerdo, e assim por diante para outros ângulos. Um equipamento define também o atributo “layer”, que representa a camada onde o elemento estará inserido no desenho, sendo que todas as camadas estarão sobrepostas, permitindo que algumas camadas do desenho sejam ocultadas, por exemplo, facilitando a visibilidade de outros elementos. Um equipamento pode ou não ter um texto associado. Em caso afirmativo, deve definir as coordenadas x (*txt_x*) e y (*txt_y*) de localização do texto, a altura (*txt_height*) do texto e sua cor (*txt_color*). As cores são determinadas de acordo com o documento XOMEF28_R01.

Cada barramento define um identificador (*id*) interno (representado por um número inteiro), o comprimento (*w*), a altura (*h*), as posições x (*x*) e y (*y*) internas (ligadas ao ponto superior esquerdo do barramento), o ângulo de rotação e a camada (*layer*). Além disso, define sua cor na tela (*color*). Cada barramento também pode definir um conjunto de pontos de ligação, com a mesma sintaxe do ponto de ligação externo (ver abaixo).

Cada aresta define os elementos ou pontos externos de origem (*source*) e destino (*target*), os pontos onde está conectada nestes elementos (*source_point* e *target_point*), sendo 1 correspondente à face direita do elemento, 2 à face superior, 3 à face esquerda e 4 à face inferior. O número de pontos de conexão depende de cada elemento. Uma aresta também define sua espessura (*w*) e cor (*color*) e pode ter um ponto de dobra, denominado “BendPoint”, o qual define as coordenadas x e y da aresta onde ocorre a quebra. Essas coordenadas são dadas em relação à origem localizada na parte superior esquerda da célula do nó.

Cada ponto de ligação externa possui um identificador (*id*) interno (número inteiro), a face (*faceID*) do nó onde está localizado (1 para face direita, 2 para superior, 3 para esquerda e 4 para inferior), e sua posição (*pos*) dentro da face, sendo sempre a origem corresponde ao canto esquerdo (faces inferior ou superior) ou superior (faces direita e esquerda).

Cada área de ligação externa possui um identificador (*id*) único (número inteiro), a face (*face ID*) onde nós serão criados (1 para face direita, 2 para superior, 3 para esquerda e 4 para inferior), o identificador do barramento que se ligará a esta área (*BusID*) e dois pontos que limitam área, o início (*bgn*) e o final (*end*), definidos por sua distância da origem da face, ou seja, seu canto esquerdo (faces inferior ou superior) ou superior (faces direita e esquerda).

Exemplo: Seja a configuração representada na Figura 1, composta de duas chaves (sendo uma delas invertida), um disjuntor e um Infeeding Line Area. Essa configuração foi extraída da subestação RCM. Ela é mapeada como um Node Template gerando a seguinte entrada no arquivo de Node Templates:

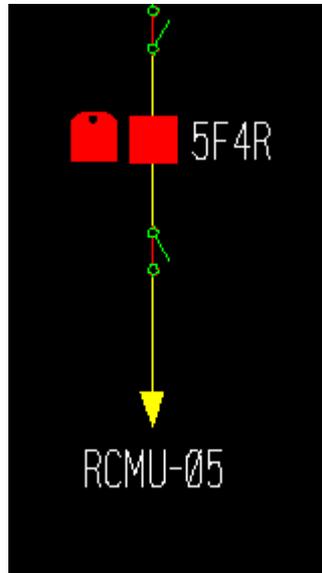


Figura 1: Node Template 2Chave_01_01_F+Disjuntor_F+IFLA da subestação RCM.

```
<Node id="2Chave_01_01_F+Disjuntor_F+IFLA" w="42.5" h="126">
```

```
  <Equipment id="1" desc="Chave_Fechada_Inv" type="Chave_01_01_F_e" w="20" h="15" x="27"
y="0" ang="270" layer="1"/>
```

```
  <Equipment id="2" desc="Chave_Aberta_Inv" type="Chave_01_01_A_e" w="20" h="15" x="27"
y="0" ang="270" layer="1"/>
```

```
  <Equipment id="3" desc="Chave_Fechada" type="Chave_01_01_F" w="20" h="15" x="27" y="56"
ang="270" layer="1"/>
```

```
  <Equipment id="4" desc="Chave_Aberta" type="Chave_01_01_A" w="20" h="15" x="27" y="56"
ang="270" layer="1"/>
```

```
  <Equipment id="5" desc="Disjuntor_Fechado" type="Disjuntor_Fechado" w="16.0" h="16.0" x="26.5"
y="30.0" ang="0" txt_x="42.5" txt_y="30" txt_height="16" txt_color="0" layer="1"/>
```

```
  <Equipment id="6" desc="Etiqueta" type="Etiqueta" w="16.5" h="18.2" x="0" y="28.9" ang="0"
layer="1"/>
```

```
  <Equipment id="7" desc="Infeeding_Line_Area" type="IFLA" w="40.0" h="10.0" x="29.5" y="86"
ang="270" layer="1"/>
```

```
  <ExternalPoint id="8" faceID="2" pos="37" />
```

```
<Edges>
```

```

<Edge source="1" target="5" source_point="1" target_point="2" w="1" color="39"/>
<Edge source="2" target="5" source_point="1" target_point="2" w="1" color="39"/>
<Edge source="5" target="3" source_point="4" target_point="3" w="1" color="39"/>
<Edge source="5" target="4" source_point="4" target_point="3" w="1" color="39"/>
<Edge source="3" target="7" source_point="1" target_point="3" w="1" color="39"/>
<Edge source="4" target="7" source_point="1" target_point="3" w="1" color="39"/>
<Edge source="8" target="1" source_point="" target_point="3" w="1" color="39"/>
<Edge source="8" target="2" source_point="" target_point="3" w="1" color="39"/>
</Edges>
</Node>

```

Note que os equipamentos que aparecem são dois pares de chave aberta/fechada ($id=1/2$ e $id=3/4$), um disjuntor ($id=5$) e um Infeeding Line Area ($id=7$). Também aparece uma etiqueta ($id=6$), que acompanha o disjuntor. O nó possui um ponto de ligação externa ($id=7$), localizado na face superior ($face/D=2$), correspondendo ao ponto superior do par de chaves superior ($id=1/2$). Os pares de chave aberta/fechada aparecem com objetivo de permitir a alternância entre os estados desligado/ligado, de acordo com o relatório da etapa 12a.

Existem quatro pares de arestas, devido à duplicação das chaves. A primeira aresta (par de arestas) liga a chave superior ao disjuntor, a segunda liga o disjuntor à chave inferior, a terceira liga a chave inferior ao Infeeding Line Area, e, por fim, a quarta liga o ponto de ligação externa à chave superior. Observe que as faces dos elementos onde as arestas estão ligadas devem sofrer uma transformação de acordo com o ângulo de rotação do elemento. Tome como exemplo a primeira aresta. O elemento de origem é a chave superior ($source=1$), rotacionada de 270 graus ($ang=270$). O ponto de origem corresponde à face 1 ($source_point=1$). Sem rotação, a face 1 corresponderia à face direita. Como foi rotacionada de 270 graus (sempre no sentido anti-horário), 1 corresponde à face inferior no desenho. O elemento de destino é o disjuntor ($target=5$), que não está rotacionado ($ang=0$). Neste caso, as faces não precisam de transformação e o ponto de destino corresponde à face 2 ($target_point=2$), que é a face superior no desenho.

4. Conclusões

O uso do programa gerado pela UFMG é extremamente simples, já que ele não se destina a ser utilizado pelo usuário final, mas sim ser chamado por outros programas (no caso o programa gerador de sistemas, da Concert) e gerar arquivos para alimentar outros programas (no caso a macro geradora de telas para o Microstation da Bentley).

5. Referências Bibliográficas

[MES08] R. C. MESQUITA e equipe da UFMG do Projeto P&D-241 - *Levantamento do estado da arte na área de desenho de grafos*, Relatório do Projeto P&D D-241 ANEEL (CEMIG/Concert/UFMG) - Desenvolvimento de Software para Conversão Automática de Diagramas de Operação de Subestações em Quadros Sinóticos e Base de Dados Digitalizada, CEMIG/UFMG, Julho de 2008.

[MEN08a] M. L. MENDES e M. H. G. FREITAS. Normatização para Confecção de Sinóticos – Especificação Funcional, XOM EF 028 R01, CONCERT/CEMIG/UFMG, Junho de 2009.