

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE
DO SUL**

**FACULDADE DE ENGENHARIA
MESTRADO EM ENGENHARIA ELÉTRICA**

**UTILIZAÇÃO DO PROTOCOLO OPC-DA NA
INTEGRAÇÃO DE UM SISTEMA OTS**

**AUTOR: TIAGO TORRES DOS SANTOS
ORIENTADOR: DR. FLÁVIO ANTONIO BECON LEMOS**

Porto Alegre, dezembro de 2008.

Agradecimentos

Agradeço aos meus pais por terem investido boa parte de suas vidas e de seus recursos financeiros em minha formação moral, ética e intelectual.

Agradeço à minha noiva por ter compreendido o tempo e a dedicação necessária para elaboração desse trabalho.

Agradeço ao Professor Dr. Flávio Antonio Becon Lemos pela orientação e incentivo.

Agradeço ao Grupo de Sistemas de Energia Elétrica da Pontifícia Universidade Católica do Rio Grande do Sul, pelas discussões e pela intensa troca de experiências de grande contribuição técnica;

Agradeço às empresas Sul Engenharia e Sistemas Ltda., Elipse Software Ltda. e PowerSysLab Engenharia e Sistemas Ltda., por fornecerem os softwares necessários para desenvolvimento desse trabalho.

Resumo

Para acompanhar a crescente demanda energética da sociedade moderna, os sistemas de geração, transmissão e distribuição de energia elétrica foram expandidos e modernizados ao longo das últimas décadas, como consequência sua complexidade operativa foi significativamente incrementada, tornando necessário o controle de um grande número de pontos de interligações e de conexões com sistemas de geração. Além disso, o processo de desregulamentação do setor elétrico fez com que as empresas utilizassem seus ativos de uma forma mais eficiente, e como consequência, mais próximo de seus limites físicos, e assim, ficando mais susceptíveis a fenômenos como perda de sincronismo, atuações de dispositivos de proteção, queda de frequência, colapso de tensão e corte de cargas. Acompanhando a complexidade operativa do sistema, um grande número de ferramentas computacionais, com diferentes funcionalidades, foram desenvolvidas especificamente para facilitar e qualificar as atividades dos centros de controle, dando origem ao que se chama hoje de EMS (*Energy Management Systems*). Diante desse contexto, surge então a necessidade da capacitação dos profissionais envolvidos na operação do sistema, uma vez que os mesmos devem ser treinados tanto no aspecto de compreensão do comportamento físico do sistema elétrico, como no aspecto de utilização das ferramentas computacionais disponibilizadas no centro de controle. Esse trabalho apresenta uma proposta de implementação de um sistema OTS (*Operator Training Simulator*) para auxiliar o treinamento de operadores de centro de controle. Para tal implementação é proposta a utilização de um protocolo industrial padronizado, o OPC-DA (*OLE for Process Control – Data Access*), que possibilita que a interface entre o sistema de simulação e o operador seja o mesmo sistema SCADA (*Supervisory Control and Data Acquisition*) utilizado para operar o sistema real, possibilitando um maior grau de proximidade entre o ambiente de treinamento e o ambiente de trabalho real. Além de possibilitar o aproveitamento dos *hardwares* e dos *softwares* SCADA já existentes no centro de controle, a abordagem proposta possibilita que diferentes tipos de treinamento sejam elaborados, disponibilizando versatilidade e agilidade ao processo de aprimoramento técnico dos profissionais envolvidos na operação do sistema elétrico.

Abstract

During the past decades the generation, transmission and distribution systems were expanded and modernized to follow the increase of the load promoted by the modern society. As a result of this growth, the power system complexity increases significantly, demanding the control of a large number of interconnections to other systems and generation units. The emergence of deregulation has dramatically changed the business environment, making the electrical utilities to maximize the use of their assets and consequently placing the system closer to its stability limits. Following the increase of the power system complexity, many computational applications were developed to help the control center operators in their tasks, creating the called EMS (Energy Management Systems). However, the operation of a modern power system has become more complicated and as a result, the need for advanced and extensive operator training becomes critically important. The operator should be trained to be familiarized with the characteristics of the system under different (and stressed) conditions and be trained with the computational tools available in the control center. This work introduces the implementation of an OTS (Operator Training Simulator) based on a standardized communication protocol called OPC-DA (*OLE for Process Control – Data Access*). The implementation of this protocol to integrate the OTS modules makes possible to build the CCM (Control Center Model) on the same SCADA (*Supervisory Control and Data Acquisition*) system used to operate a real power system. The solution proposed allows the existent control center SCADA *hardware* and software be used, saving resources and making the training environment very similar to the real one. Different kinds of training are explored through the OTS proposed.

Sumário

1.	Introdução.....	9
1.1.	Revisão Bibliográfica.....	12
1.2.	Contribuição da Dissertação.....	15
1.3.	Estrutura da Dissertação.....	16
2.	Simulador de Treinamento de Operadores (OTS).....	18
2.1.	Funções Específicas de um OTS.....	19
2.1.1.	Funções SCADA.....	19
2.1.2.	Funções de Regime Permanente.....	19
2.1.3.	Funções Dinâmicas.....	20
2.1.4.	Funções de Gerenciamento de Energia.....	20
2.2.	Módulos do OTS.....	20
2.2.1.	PSM – Power System Model.....	21
2.2.2.	ES – Educational System.....	24
2.2.3.	CCM – Control Center Model.....	25
2.3.	Fluxo de Informações entre os Módulos do OTS.....	26
2.4.	Tipos de Simuladores de Treinamento.....	28
2.4.1.	Stand-Alone.....	29
2.4.2.	Integrado as Aplicações do Centro de Controle.....	30
3.	OPC - Object Linking and Embedding for Process Control.....	32
3.1.	DCOM.....	34
3.2.	OPC Overview/OPC Common Definition and Interfaces.....	35
3.3.	OPC Data Access.....	36
3.4.	OPC Alarms and Events.....	38
3.5.	OPC Historical Data Access.....	39
3.6.	OPC XML-DA.....	40
4.	Construção dos Módulos PSM e ES.....	43
4.1.	FASEE: Framework para Análise de Sistemas de Energia Elétrica.....	43
4.2.	Ferramenta de Modelagem e Construção de Ambientes de Simulação de Sistemas Elétricos.....	45
4.3.	Módulo ES sobre o PSL®-SDK.....	46
4.4.	Adaptação de um Servidor OPC ao PSL-SDK.....	47
4.5.	Modelagem de um Sistema Exemplo sobre o FASEE.....	50
5.	Construção do CCM sobre um Software SCADA.....	55
5.1.	Sistemas SCADA.....	56
5.2.	Elipse E3.....	58
5.2.1.	Drivers de Comunicação.....	58
5.2.2.	Alarmes.....	59
5.2.3.	Eventos.....	59
5.2.4.	Histórico.....	60
5.2.5.	Estrutura Cliente Servidor.....	60
5.2.6.	Biblioteca de Usuário – ElipseX.....	61

5.3.	Software S4	62
5.3.1.	Estruturação dos Drivers de Comunicação	62
5.3.2.	Estruturação da Camada de Dados	64
5.3.3.	Telas de Operação do Sistema Elétrico	65
5.3.4.	Construção do Sistema Exemplo	65
5.4.	Adaptação do Cliente OPC ao módulo SCADA/CCM	67
6.	Integração dos Módulos ES, PSM, SCADA e CCM	74
6.1.	Possíveis Arquiteturas para Distribuição dos Processos em Rede	74
6.1.1.	Stand-Alone	74
6.1.2.	Distribuído em dois Servidores e Diversos Clientes	75
6.1.3.	Distribuído em Diversos Servidores e Diversos Clientes	76
6.2.	Estrutura de Rede Utilizada para Testes e Avaliação	77
6.3.	Configuração do Microsoft Windows XP Service Pack 2 para Liberação do OPC/DCOM	78
6.3.1.	Configuração do Windows Firewall	78
6.3.2.	Configuração do DCOM	79
6.4.	Operação Integrada dos Módulos	80
7.	Avaliação e Interpretação dos Resultados	81
7.1.	Modo Stand-Alone	81
7.2.	Treinamento Supervisionado	82
7.3.	Simulação de Ações em Tempo Real	83
8.	Conclusão	84
8.1.	Perspectivas Futuras	86
	Referências Bibliográficas	88

Lista de Figuras

Figura 1: Fluxo de informações de um OTS.	26
Figura 2: Ambiente OTS na forma Stand-Alone.	29
Figura 3: Especificações OPC e suas Inter-relações.	34
Figura 4: <i>Namespace</i> e Hierarquia de Objetos em um Servidor Data Access.	37
Figura 5: <i>Namespace</i> e Hierarquia de Objetos no Historical Data Server.	40
Figura 6: Integração de informações disponibilizadas por servidores OPC-DA para a Internet via OPC XML-DA.	42
Figura 7: Curva de carga criada através do módulo ES do PSL [®] -SDK.	47
Figura 8: Hierarquia dos dados internos no servidor OPC do PSM.	49
Figura 9: Tela de representação do sistema exemplo implementado no módulo PSM.	50
Figura 10: Tela de implementação do modelo da subestação 4 do sistema exemplo.	51
Figura 11: Tela de criação de eventos de abertura de chaves.	52
Figura 12 – Tela do OPC Explorer utilizado para consultar aos valores simulados pelo módulo PSM.	53
Figura 13: Diagrama esquemático de um sistema SCADA.	57
Figura 14: Ambiente de parametrização de um driver de comunicação.	63
Figura 15: Estruturação dos objetos de dados na aplicação S4.	64
Figura 16: Representação Gráfica de um transformador no S4.	65
Figura 17: Implementação do sistema exemplo de 45 barras sobre o S4.	66
Figura 18: Pontos OPC criados em um cliente dentro do sistema SCADA.	68
Figura 19: Propriedades de um objeto de dado digital apontando para um tag de comunicação proveniente de uma UTR.	69
Figura 20: Propriedades de um objeto de dado digital apontando para um tag de comunicação proveniente do módulo PSM.	70
Figura 21: Propriedades de um objeto de dados responsável por modelar um módulo de alimentador contendo informações reais.	71
Figura 22: Propriedades de um objeto de dados responsável por modelar um módulo de alimentador contendo informações simuladas.	71
Figura 23: Diagrama de operação de uma subestação monitorada por uma UTR.	72
Figura 24: Diagrama de operação de uma subestação simulada pelo módulo PSM e acessada via CCM.	73
Figura 25: Plataforma configurada para trabalhar no modo Stand-Alone.	75
Figura 26: Plataforma configurada para trabalhar distribuída em dois servidores.	76
Figura 27: Plataforma configurada para trabalhar distribuída em três servidores.	77
Figura 28: Inclusão de Exceções no Windows Firewall.	79
Figura 29: Tela de configuração de serviços DCOM.	80

Lista de Abreviações

ANEEL	– Agência Nacional de Energia Elétrica.....	10-11
API	- Application Program Interface.....	87-88
CAG	- Controle Automático da Geração.....	13, 26
CCAPI	- EPRI Control Center API	87-88
CCEE	- Câmara de Comercialização de Energia Elétrica	10
CCM	- Control Center Model	24, 73
CLP	- Controlador Lógico Programável.....	33, 39, 59
CLSID	- Class Identifier	36
CMSE	- Comitê de Monitoramento do Setor Elétrico	10
COS	- Centro de Operação de Sistema Elétrico	56, 63
CPST	- Contratos de Prestação do Serviço de Transmissão	10
CPU	- Central Process Unit	33
DCOM	- Distributed Component Object Model	15, 34, 79
DLL	- Dynamic Link Library	36, 59
DTS	- Dispatcher Training Simulator	11, 14, 19
EMS	- Energy Management Systems	11-16, 26, 30, 87
EPE	- Empresa de Pesquisa Energética	10
EPRI	- Electric Power Research Institute	14, 87
ERP	- Enterprise Resource Planning	34
ES	- Educational System	12, 17, 21-28, 34, 44-48, 75
HTML	- Hypertext Markup Language	41
IDL	- Interface Definition Language	36
IHM	- Interface Homem-Máquina	57
IS	- Instructor Position	21
MAE	- Mercado Atacadista de Energia	10
MES	- Manufacturing Execution Systems	34
OLE	- Object Linking and Embedding	15, 34
ONS	– Operador Nacional do Sistema	10
OPC	- OLE for Process Control	31-39, 46
OPC-DA	- OLE for Process Control – Data Access	35
OTS	- Operator Training Simulator	17-30
PC	- Personal Computer	37
PSM	- Power System Model	20-23, 42-49
SCADA	- Supervisory Control and Data Acquisition	18, 54-60
SIN	- Sistema Interligado Nacional	10
SQL	- Structured Query Language	47, 59
UTR	- Unidade Terminal Remota	63, 70
XML	- eXtensible Markup Language	41-48

1. Introdução

Os sistemas de energia elétrica sofreram grandes mudanças a partir da década de 80 do século XX, passando de uma estrutura verticalizada e de característica fortemente centralizada e estatal para um modelo onde as atividades de transmissão permaneceram sob o controle estatal e a geração e distribuição passaram a ser exploradas por empresas privadas e estatais. No Brasil, esta situação criou um ambiente para a competição que necessitou ser novamente regulamentado por um novo conjunto de Leis, Decretos e Resoluções [1]. Surgiu então, como consequência deste novo modelo, dois agentes: a ANEEL – Agência Nacional de Energia Elétrica [1], cuja missão é proporcionar condições favoráveis, através da regulamentação e fiscalização para que o mercado de energia elétrica se desenvolva com equilíbrio entre os agentes e em benefício da sociedade, o ONS – Operador Nacional do Sistema [2], responsável pela coordenação e controle da operação das instalações de geração e transmissão de energia elétrica no Sistema Interligado Nacional (SIN). Durante os anos de 2003 e 2004 o Governo Federal lançou as bases de um novo modelo para o Setor Elétrico Brasileiro, sustentado pelas Leis nº 10.847 e 10.848, de 15 de março de 2004; e pelo Decreto nº 5.163, de 30 de julho de 2004 [1]. Em termos institucionais, o novo modelo definiu a criação de uma entidade responsável pelo planejamento do setor elétrico em longo prazo (a Empresa de Pesquisa Energética – EPE), uma instituição com a função de avaliar permanentemente a segurança do suprimento de energia elétrica (o Comitê de Monitoramento do Setor Elétrico – CMSE) e uma instituição para dar continuidade às atividades do MAE (Mercado Atacadista de Energia), relativas à comercialização de energia elétrica no Sistema Interligado (a Câmara de Comercialização de Energia Elétrica - CCEE) [3].

A legislação atual do setor elétrico prevê nos Contratos de Prestação do Serviço de Transmissão - CPST, um mecanismo de penalidade denominado Parcela Variável – PV, onde a indisponibilidade dos ativos de transmissão, programada ou não, é onerada por um desconto sobre sua receita. Este mecanismo foi criado com o objetivo de incentivar (maximizar) a disponibilidade dos equipamentos, instalações e serviços prestados pelas empresas detentoras de concessão do serviço de transmissão de energia. Até junho de 2008 a legislação que penalizava a

indisponibilidade dos ativos de transmissão era aplicada somente a novas instalações licitadas. Entretanto, a ANEEL, através da Resolução Normativa nº270, de 26/06/2007, a qual estabelece disposições relativas à qualidade do serviço público de transmissão de energia elétrica, estendeu a aplicação da parcela variável aos ativos pertencentes a todas as instalações das transmissoras (antigas e novas) a partir de junho de 2008.

Entre outras regras estabelecidas nesta Resolução consta a que estabelece que o desconto da Parcela Variável devido à indisponibilidade do equipamento de forma não programada pode ser até 15 vezes maior que seria no caso do desligamento ter sido programado com antecedência. Dessa forma, a indisponibilidade de equipamentos que conduzam a desligamentos forçados de linhas de transmissão com duração superior a um minuto tornam-se passíveis de aplicação da PV. O cenário criado por esta regulamentação torna fundamental a avaliação dos processos envolvidos nas ações de proteção e controle que podem levar a indisponibilidade dos ativos dos sistemas de transmissão.

Uma das formas de melhorar o desempenho dos sistemas de transmissão, que englobam subestações, linhas de transmissão, equipamentos e dispositivos de controle e proteção associados é investir em automação. Nos últimos anos, devido à diminuição dos custos de equipamentos e ao surgimento de novas tecnologias, o processo de automação de diversas atividades associadas ao setor elétrico passou a ser fundamental e tornou-se realidade em todas as empresas de energia elétrica. Esta situação aumentou a quantidade de informações e opções de ações de controle, tornando a operação do sistema mais complexa e “impondo pressão” sobre os operadores e exigindo tomada de decisões mais ágeis. Sendo assim, espera-se que os operadores de sistemas elétricos estejam familiarizados com o sistema de energia sobre o qual irão tomar ações de controle em condições normais e durante situações de emergência, de forma a maximizar a segurança do sistema e evitar penalidades regulatórias.

Esses fatores tornam evidente a necessidade de familiarizar e treinar futuros operadores para executar ações sobre o sistema real, bem como reciclar operadores com novos procedimentos criados para melhorar a segurança operativa, principalmente nas fases de emergência e recomposição. O conceito de treinamentos baseados em simulações, solução que já era comum em diferentes setores da indústria, foi introduzido no setor elétrico em 1977 no formato de um DTS (*Dispatcher Training Simulator*) [4]. Este tipo de simulador também é referenciado pela sigla inglesa OTS – *Operator Training Simulator* [5].

Filosoficamente, o OTS faz parte dos sistemas de Energy Management System – EMS [6], sendo utilizado para reproduzir o modelo elétrico e simular o comportamento de um sistema de energia em situação normal ou quando submetido a alterações nas condições operacionais,

causadas por contingências pré programadas ou impostas por um instrutor. A principal função do OTS esta em recalcular valores de tensão e potência toda vez que algum evento ou alteração de parâmetro ocorrer, e apresentar estas informações em um ambiente que reproduza o Centro de Operação. Isto permite que o operador em treinamento esteja inserido em um ambiente que reproduz situações que ocorrem durante a operação do sistema, permitindo ao instrutor monitorar as reações do operador em treinamento, quando seguindo ordens de operação, bem como as habilidades do operador em se antecipar a futuros eventos e sua velocidade de reação quando submetido a um estresse de ações imposta pelo instrutor [7].

Os principais requisitos para desenvolver e introduzir um OTS para treinamento são [6]:

- Desenvolver um conjunto de cenários para situações corriqueiramente encontradas durante a operação normal de um sistema elétrico, bem como possíveis cenários compostos de listas de contingências para gerar uma operação em condições de emergência;
- Criar um ambiente de treinamento o mais próximo possível do Centro de Controle que está sendo reproduzido pelo OTS;
- Disponibilizar um conjunto suficiente de informações de tempo-real capaz de reproduzir todos os eventos possíveis de acontecerem na operação;
- Permitir facilidade de customização de diferentes tipos de treinamento, tais como auto-treinamento e treinamento supervisionado;
- Facilitar a inserção de novas funções de simulação para expandir o OTS;
- Suportar e auxiliar o desenvolvimento de metodologias para detecção de vulnerabilidades no sistema elétrico que está sendo simulado;
- Auxiliar no comissionamento de centros de operação.

Basicamente um OTS é composto pelos seguintes módulos:

1. CCM (*Control Center Model*) o qual reproduz o ambiente do centro de operação do sistema
2. PSM (*Power System Model*) responsável por simular os componentes da rede elétrica e a resposta dinâmica dos equipamentos do sistema
3. ES (*Educational System*) também é conhecido como *Construtor e Avaliador de Treinamento*, é o módulo onde o instrutor carrega as condições de operação da

rede e o ponto de operação inicial. Este módulo também é responsável pela avaliação durante e ao final da seção de treinamento.

Nessa dissertação é apresentada uma proposta de integração entre os módulos CCM e PSM, utilizando um protocolo padrão de comunicação de dados, o OPC-DA, que possibilita que o módulo CCM seja desenvolvido sobre a mesma plataforma SCADA utilizada no centro de controle para operar o sistema real.

1.1. Revisão Bibliográfica

Sistemas de geração, transmissão e distribuição de energia elétrica são supervisionados por diversos centros de controle, os quais atuam sobre diferentes regiões possuindo diferentes níveis de hierarquia e complexidade. Com o desenvolvimento de tal sistema, o aumento de suas interligações e automação dos centros de controle, o trabalho dos operadores atingiu um grau de complexidade e responsabilidade consideravelmente maior do que o vigente há alguns anos atrás [8] tornando o centro de controle da empresa que trabalha com energia elétrica, o principal agente para garantir a segurança operativa do sistema [9]. Com o sistema sendo operado perto de seus limites físicos, seu funcionamento está cada vez mais sujeito a fenômenos como perda de sincronismo, atuação de dispositivos de proteção, queda de frequência, colapso de tensão, corte de geradores e cargas [7].

Acompanhando o aumento da complexidade das atividades executadas pelos operadores nos centros de controle, uma série de aplicativos computacionais chamados de EMS (*Energy Management Systems*) foram desenvolvidos ao longo dos últimos anos [10]. A primeira geração desse tipo de aplicativo, os chamados sistemas SCADA (*Supervisory Control and Data Acquisition*) eram baseados em computadores e sistemas operacionais especiais [11]. Estes sistemas normalmente realizavam apenas funções básicas como coleta e processamento de dados de tempo real. No início dos anos 80, esses sistemas evoluíram, integrando funções de aplicação como CAG (Controle Automático da Geração). No início dos anos 90, tais sistemas começaram a ser substituídos por sistemas baseados em Unix. Esses sistemas se beneficiaram de novas tecnologias como bases de dados relacionais e recursos gráficos sofisticados, apresentando

excelente estabilidade quando comparados aos sistemas anteriores. Durante essa década também surgiram algumas propostas para que houvesse uma forma padronizada e aberta que permitisse a integração entre as diferentes aplicações que constituem o EMS, permitindo assim que aplicações de diferentes fornecedores trabalhassem de forma integrada [12].

Diante do crescimento da complexidade operativa dos sistemas elétricos e do número de sofisticadas aplicações computacionais disponibilizadas ao operador dentro do centro de controle, surgiu a necessidade da constante capacitação desses profissionais, de forma que os mesmos estejam sempre aptos a responder de forma rápida e eficiente [13]. Usualmente os operadores do sistema são treinados através de aulas teóricas e durante a operação do sistema real. Como o sistema elétrico está constantemente crescendo em tamanho e complexidade, a capacitação do operador deve acompanhar esse crescimento. Novos operadores se deparam com o desafio de assimilarem o conhecimento de operadores mais experientes através de relatórios e do compartilhamento de experiências dentro do ambiente de trabalho. Esse tipo de treinamento não se mostra eficiente para preparar o operador a uma grande gama de situações que podem ocorrer durante um distúrbio no sistema elétrico. Historicamente, a maioria dos problemas que ocasionaram situações de emergência, quase *Blackout*, ou *Blackout*, foram ocasionados por problemas não vivenciados anteriormente [14]. Todos esses fatores contribuíram para a evidência da necessidade de um sistema de treinamento de operadores de centros de controle em todas as fases da operação: normal, emergência e de recomposição. O conceito de treinamento baseado em simulação, solução que já era comum em diferentes setores da indústria e foi introduzido no setor elétrico em 1977 no formato de um DTS [4]. Esse DTS possuía capacidades limitadas em termos de tamanho de modelo, de integridade de resultados de simulação, interface de usuário e funções de controle. O assunto começou a ter uma atenção especial tanto da academia quanto da indústria, pois esforços em pesquisa eram necessários para que os simuladores atendessem de forma satisfatória as necessidades dos centros de controle. Nos anos seguintes, muitos vendedores introduziram sistemas DTSs em seus EMSs. O tamanho dos modelos cresceram com o passar do tempo e os resultados das simulações se mostraram mais fiéis a realidade. Dentre as diversas pesquisas e desenvolvimentos no assunto, pode-se ressaltar o trabalho desenvolvido pelo EPRI (*Electric Power Research Institute*) que é uma organização sem fins lucrativos, independente, que reúne profissionais experientes da academia e da indústria, com a finalidade de ajudar a desenvolver soluções nas áreas de geração, transmissão e utilização de energia elétrica [5]. Porém, devido a fatores como: abrangência do assunto,

diferentes tipos de simulações e diferentes necessidades das empresas, atualmente diversas propostas de simuladores de treinamento de operadores podem ser encontradas na literatura [15], [16], [17] e [18].

Porém, com o aumento do número de fornecedores e aplicações disponibilizadas no mercado ao longo das três últimas décadas, diversos problemas de integração impactaram no desempenho dos EMSs e consecutivamente dos sistemas de OTS. Os sistemas a serem integrados mostraram-se pobres em disponibilizar interfaces abertas, suas arquiteturas não suportavam as demandas das empresas que presenciavam um rápido crescimento do número de consumidores e consecutivamente, o crescimento do sistema elétrico, tornando a manutenção e ampliação do sistema uma tarefa cada vez mais árdua. Diante desse cenário, novas formas de implementação do ambiente de EMS foram propostas com a finalidade de padronizar o intercâmbio de dados e as interfaces entre os diferentes aplicativos que constituem esse sistema, como em [19], [20], [21], [22], [23] e [24].

A experiência negativa vivenciada pelo setor elétrico também ocorreu em outros setores da indústria, onde um grande número de soluções de automação também utilizava sistemas proprietários, tornando praticamente impossível a integração de equipamentos e softwares de diferentes fabricantes. Com o aumento da competitividade e da demanda da indústria por sistemas cada vez mais abertos, eficientes e de baixo custo, o surgimento de um padrão que fosse obedecido por um grande número de desenvolvedores de sistemas de automação era evidente. Diante dessa demanda, uma das soluções propostas pelo mercado de automação veio através de um consórcio formado por usuários e desenvolvedores de soluções de automação, a *OPC Foundation* [25], a qual conta com algumas empresas fornecedoras de soluções para o setor elétrico como Siemens, ABB e GE [26], e tem como finalidade, estudar formas de facilitar a integração entre as diferentes fontes de dados comuns em plantas industriais altamente automatizadas [27]. Como resultado dos estudos desenvolvidos por essa fundação, surgiu o OPC (*OLE for Process Control*), que é um protocolo de comunicação baseado em uma série de padronizações OLE/DCOM (*Object Linking and Embedding / Distributed Component Object Model*), as quais foram desenvolvidas pela Microsoft para facilitar e efficientizar o processo de desenvolvimento e integração de sistemas [28] e o reaproveitamento de código, introduzindo o conceito de componentes [29]. O OPC é o primeiro componente padronizado desenvolvido para atender as necessidades específicas do domínio da automação, padronizando os mecanismos para comunicar com inúmeras fontes de dados como equipamentos de campo e bancos de dados em

centros de controle. Dessa forma, os fabricantes de equipamentos de automação podem concentrar esforços no desempenho da comunicação com suas fontes de dados locais e padronizar o mecanismo de acesso de sistemas externos a esses dados. As interfaces OPC disponibilizadas pelos servidores permitem que qualquer cliente acesse seus dispositivos internos de forma padronizada [30].

O desempenho do OPC foi considerado em [31] adequado para a maioria das aplicações dedicadas e distribuídas que são executadas sobre plataformas computacionais comerciais.

A utilização do OPC vem sendo uma alternativa comumente utilizada para integrar dados de diferentes aplicações em centros de controle de sistemas de energia elétrica. Alguns trabalhos foram realizados para integração de informações em centros de controle, [32], [33], [34] e [35]. Devido à versatilidade, facilidade e quantidade de usuários dessa tecnologia, o padrão de integração de informações de sistemas SCADAs (*Supervisory Control and Data Acquisition*) e de aquisição de dados, chamado de IEC 61970 part 303, leva em consideração a utilização do OPC [36]. Algumas soluções também foram implementadas para integrar as informações de simuladores utilizando esse protocolo, como em [37] e [38], porém nesses casos os simuladores não são destinados para treinamento de operadores de sistemas elétricos.

1.2. Contribuição da Dissertação

As principais contribuições do trabalho de pesquisa que originou essa dissertação podem ser sintetizadas em:

- Propõe a utilização do protocolo OPC-DA como meio de interconexão entre os módulos CCM e PSM de um OTS;
- Introduz o conceito que um grande número de sistemas SCADAs comerciais possam integrar as informações dos módulos PSMs à sua estrutura de dados interna, havendo maior integração entre as aplicações que constituem um EMS.
- Apresenta as funcionalidades e modos de treinamento que o sistema OTS pode implementar devido versatilidade, desempenho e padronização do OPC-DA;
- Propõe uma forma de estruturação de dados no servidor OPC-DA que permita a identificação da semântica de cada ponto de comunicação;

- Identifica as principais deficiências que o sistema OTS pode apresentar devido à utilização do protocolo e propõe algumas soluções que tornariam o sistema mais robusto e de mais fácil implantação.

Esse trabalho também resultou uma publicação e apresentação em congresso internacional.

“Utilização do OPC para Integrar Sistemas SCADA a Sistemas de OTS”, T. T. dos Santos, F. A. B. Lemos, A. Manzoni, A. V. Zampieri, Décimo Segundo Encontro Regional Ibero-Americano do CIGRÉ, Foz do Iguaçu-PR, Brasil, 20-24 de maio de 2007.

1.3. Estrutura da Dissertação

Esta dissertação está estruturada da seguinte forma:

Capítulo 1: É apresentada uma breve introdução sobre a operação de sistemas elétricos, sobre os aplicativos computacionais existentes nos centros de controle e sobre as necessidades de integração entre essas aplicações, ressaltando-se a importância do investimento no aperfeiçoamento técnico dos profissionais envolvidos na operação do sistema. Para isto é realizada uma revisão bibliográfica sobre o assunto, de forma a identificar algumas soluções aos problemas existentes.

Capítulo 2: Apresenta uma conceituação do termo OTS, identificando os principais módulos que o constituem, suas variações de implementação e as principais funcionalidades desse sistema. Nesse capítulo também é justificada a necessidade de implantação desse sistema em um centro de controle.

Capítulo 3: Apresenta o protocolo padrão de comunicação de dados OPC e suas diferentes versões, abordando de forma sucinta a tecnologia utilizada pelo protocolo, seu objetivo e suas características funcionais.

Capítulo 4: Neste capítulo é apresentada a ferramenta computacional utilizada para realizar a função do módulo PSM e ES no ambiente OTS proposto neste trabalho. Também é abordada a adaptação de um módulo servidor de comunicação OPC-DA para disponibilizar as informações simuladas pelo PSM ao módulo CCM.

Capítulo 5: Nesse capítulo é apresentado o modelo de CCM (*Control Center Model*) desenvolvido para ser integrado ao OTS proposto nesse trabalho. Apresenta de forma detalhada a implementação do módulo CCM sobre uma plataforma SCADA comercial.

Capítulo 6: Esse capítulo apresenta a integração entre os módulos CCM e PSM, bem como as diferentes arquiteturas que podem ser implementadas pelo OTS proposto. São abordados aspectos de configuração e parametrização de componentes OPC/DCOM e firewalls.

Capítulo 7: Nesse capítulo são avaliados e interpretados os resultados obtidos, ressaltando os pontos positivos e negativos da estrutura proposta.

Capítulo 8: Apresenta a conclusão do trabalho desenvolvido e as perspectivas de estudos futuros.

2. Simulador de Treinamento de Operadores (OTS)

O simulador de treinamento de operadores, conhecido pela sigla inglesa OTS (*Operator Training Simulator*), e algumas vezes também referenciada como DTS (*Dispatcher Training Simulator*) [4], é um sistema computacional utilizado para reproduzir o modelo elétrico e simular o comportamento de um sistema de energia em situação normal ou quando submetido a alterações nas condições operacionais, causadas por contingências pré-programadas ou impostas por um instrutor. Essa ferramenta permite que o treinamento de novos operadores seja realizado de forma segura e em um curto prazo [8]. A principal função do OTS está em recalculando valores de tensão e potência toda vez que algum evento ou alteração de parâmetro ocorrer [13]. Isso permite analisar, de forma segura o comportamento que o sistema real teria quando submetido a essas alterações. De forma similar, quando o operador responde ao evento ou a uma condição especial através de comandos de abertura / fechamento de disjuntores, ajuste da posição de tap de transformadores, etc., o simulador recalcula o estado do sistema, apresentando uma nova condição operativa.

Como geralmente o tempo de resposta do simulador é muito parecido com o tempo de varredura de sistemas SCADA (*Supervisory Control and Data Acquisition*) [5], os operadores sob treinamento podem receber a ocorrência dos eventos da mesma forma verificada na console de operação do sistema real. Para realizar estas tarefas, o OTS deve disponibilizar uma interface similar à utilizada na operação do sistema real, a fim de fornecer um ambiente o mais próximo possível do ambiente do centro de operação. Dessa forma, os operadores podem adquirir experiência e serem treinados para lidar com um grande número de situações normais e de emergência que se assemelham as encontradas no dia a dia da operação.

2.1. Funções Específicas de um OTS

Do ponto de vista de requisitos de modelagem, as principais funções de treinamento de um sistema OTS são [6]:

2.1.1. Funções SCADA

O treinamento nesta classe inclui as funções de supervisão e controle, tais como:

- Operações normais de chaveamento;
- Controle estático de tensão/ potência reativa;
- Avaliação de limites operacionais;
- Falhas de equipamentos de transmissão de dados;
- Envio de sinais de atuação (*trip*) para disjuntores e transformadores (tap);
- Envio de sinais de mudança de geração.

2.1.2. Funções de Regime Permanente

Funções de treinamento desta classe estão relacionadas à avaliação do carregamento instantâneo do sistema e ao controle da potência gerada, avaliando requisitos de segurança estática e fatores energéticos, apresentando as seguintes funções:

- Monitoração topológica;
- Estimacão de estados;
- Fluxo de potência;
- Análise de contingências;
- Simulação de faltas (curto-circuito e perda de fase);
- Despacho econômico de carga;
- Programação de intercâmbio;
- Fluxo de potência ótimo;
- Gerenciamento de carga.

2.1.3. Funções Dinâmicas

Esta classe de funções de treinamento requer um conjunto sofisticado de modelos de equipamentos que compõe o sistema elétrico, tais como geradores e seus controladores, cargas dinâmicas, etc. Quando um OTS implementa esta classe de funções de treinamento os algoritmos são mais complexos e exigem maior esforço computacional. Dentre as funções incorporadas por esta classe estão:

- Controle automático da geração (carga-frequência);
- Controle de tensão/potência reativa;
- Avaliação de separação de sistemas e ilhamento;
- Corte de carga;
- Redespacho da geração para recomposição do sistema.

2.1.4. Funções de Gerenciamento de Energia

Esta classe de funções de treinamento está ligada a programação energética, principalmente no que se refere às previsões de curto e médio prazo. As seguintes funções são exemplos de tarefas abrangidas por este tipo de treinamento:

- Previsão de carga;
- Alocação de unidades;
- Avaliação de programas de intercâmbio;
- Negociação de compra e venda de energia.

Nesse trabalho serão utilizadas apenas as funções SCADA e de Regime Permanente, não sendo abrangidas as funções Dinâmicas e de Gerenciamento de Energia.

2.2. Módulos do OTS

Basicamente, o OTS é composto por três subsistemas ou módulos [5], são eles: o CCM (*Control Center Model*), o PSM (*Power System Model*) e o ES (*Educational System*). Em alguns casos o módulo de ES pode gerar um quarto módulo chamado de IS (*Instructor Position*) [14],

porém nesse trabalho serão considerados apenas três módulos, sendo o IS parte integrante do ES. A descrição de cada módulo é apresentada a seguir.

2.2.1. PSM – Power System Model

Este módulo costuma também ser referenciado como Simulador do Sistema Elétrico, o qual é responsável por simular os componentes da rede elétrica e a resposta dinâmica dos equipamentos do sistema. Através desse módulo os comportamentos estáticos e dinâmicos dos geradores, linhas de transmissão, transformadores, sistemas de proteção, cargas, entre outros, são simulados e enviados ao módulo CCM. O PSM é responsável por carregar a configuração inicial do sistema, fornecida pelo construtor de cenários (módulo ES), e recalcular o estado das medidas de supervisão conforme a variação dos parâmetros operativos do sistema. Geralmente, pretende-se que as informações sejam recalculadas no mesmo tempo de varredura do sistema SCADA. O objetivo do PSM é disponibilizar as mesmas informações fornecidas pelas Unidades Terminais Remotas (UTRs) existentes no sistema elétrico. Para realizar essa tarefa, o simulador deve possuir o modelo matemático de cada elemento do sistema elétrico e o cadastro de todos os pontos supervisionados do sistema. Basicamente o PSM deve implementar:

- **Simulação de eventos estáticos e dinâmicos:** O PSM é responsável por simular o comportamento estático e a parte de dinâmica lenta do sistema de potência. Com relação à simulação estática, pode-se utilizar o exemplo da abertura de um disjuntor. O simulador do comportamento estático irá determinar o estado do sistema alguns segundos após a abertura do disjuntor, devolvendo os valores esperados após o sistema estabilizar, não considerando o caminho traçado pelas variáveis para chegar a esses valores finais. Já com relação à simulação da dinâmica lenta do sistema, o PSM simula as variações de alguns valores durante alguns segundos após o sistema ter sua configuração alterada. Um exemplo típico que necessita de uma simulação de dinâmica lenta é a retirada de um gerador do sistema. Esse tipo de evento faz com que os outros geradores conectados ao sistema tenham que aumentar suas potências geradas, podendo ocasionar variação na frequência do sistema. A simulação dinâmica pode ser mantida durante alguns segundos após a ocorrência do evento. O PSM não simula os eventos de dinâmica rápida ou transitórios que ocorram em um curto intervalo de tempo (alguns milissegundos). Um exemplo desse tipo de evento pode ser o comportamento da tensão após a saída de operação de uma linha de transmissão ou a variação de potência no sistema quando

submetido a um forte distúrbio. O fato do PSM não simular o transitório não acarreta uma grande perda de funcionalidade do OTS, pois quando o operador opera o sistema real o SCADA geralmente atualiza os valores coletados em um intervalo de aproximadamente um segundo. Isso significa que esses eventos geralmente não são analisados em tempo real. Porém as conseqüências provocadas por transitórios podem ser severas. Nesses casos, a atuação de um relé de proteção pode acarretar, por exemplo, na abertura de um disjuntor de linha. Essas conseqüências não serão simuladas pelo OTS e devem ser inseridas pelo instrutor em tempo de treinamento.

- **Simulação de sistemas internos e externos:** O PSM possibilita um modelo completo do sistema sob análise (sistema interno) e um modelo simplificado ou equivalente do sistema externo [39]. O escopo e o nível de detalhamento do modelo interno dependem da necessidade do usuário e do esforço necessário para construir e manter a base de dados de cadastro do sistema. Simuladores desenvolvidos para simular regiões inteiras de transmissão costumam modelar todo sistema de transmissão e as partes mais críticas dos sistemas de subtransmissão. Já os simuladores desenvolvidos para simular sistemas de subtransmissão geralmente elaboram um modelo detalhado do sistema de subtransmissão e modelos simplificados nos pontos onde exista conexão com sistemas externos. As áreas de controle e os sistemas de empresas adjacentes podem ser modelados como um sistema externo. O modelo externo pode incluir curvas de cargas simuladas, operação do AGC, geração equivalente, etc.
- **Modelos de carga:** O modelo da carga é uma característica importante do OTS [4]. Mesmo que não ocorra nenhuma operação de disjuntor ou qualquer outro evento no sistema, a variação de carga durante um dia faz com que diferentes valores de fluxo e tensão sejam verificados. Dois aspectos importantes de serem verificados a respeito do modelo de carga no OTS: a alocação dos elementos de carga em linhas de transmissão e alimentadores e a variação da mesma durante um intervalo de simulação. Devem-se considerar diferentes curvas de carga, pois a mesma varia devido à época do ano e do dia da semana que se deseja simular.
- **Modelos dos equipamentos do sistema:** O PSM é responsável por modelar o comportamento de diferentes tipos de equipamentos presentes em sistemas de potência. Isso inclui tanto aqueles cuja influencia no sistema de potência é constante, quanto aqueles cujos efeitos variam no decorrer do tempo, como no caso de transformadores

com ajuste automático da posição do tap. Modelos de linhas de transmissão, subestações, equipamentos de manobra, capacitores, reatores, compensadores estáticos de reativo e transformadores são essências para que as respostas obtidas pelas simulações se aproximem às apresentadas pelo sistema de potência real.

- **Modelos de geradores:** O modelo do gerador deve ser capaz de simular o comportamento temporal da potência entregue pela unidade e possibilitar que a tensão na barra de saída da unidade seja controlada de forma automática, ou através da regulação manual do valor de excitação do gerador. Outra característica importante no modelo do gerador é a curva de capacidade, a qual estabelece os limites de operação de potência ativa e reativa do equipamento.
- **Modelos dos relés de proteção:** Os relés de proteção atuam sobre o sistema elétrico quando algum limite de proteção configurado for violado. Um exemplo pode ser a abertura de um disjuntor de linha quando detectada a ocorrência de sobre-corrente. Do ponto de vista do OTS, esse efeito pode ser replicado através de diferentes estratégias, podendo variar de acordo com a necessidade de cada empresa e o grau de complexidade do PSM. Algumas estratégias são:
 - **Modelar o efeito de cada relé de proteção:** Essa estratégia geralmente modela os relés 32 (Relé Direcional de Potência), 81 (Relé de Sub/Sobre Frequência), 27 (Relé de Subtensão), 59 (Relé de Sobretensão), 79 (Relé de Religamento), 25 (Relé de Verificação de Sincronismo), 62 (Relé Temporizador) e 24 (Relé de Sobreexcitação). A desvantagem dessa estratégia está na complexidade de modelar todas as proteções existentes no sistema e de manter o cadastro de proteção atualizado no módulo PSM.
 - **Construir eventos manualmente:** Nessa abordagem, o instrutor constrói scripts de forma a replicar o comportamento da atuação da proteção (Ex.: Se a tensão ultrapassar 360 kV, Abrir Banco de Capacitores).
 - **Construir grupo de eventos:** Essa implementação possibilita que faltas sejam replicadas através de um grupo de eventos onde são consideradas operações de disjuntores e alarmes que devem ser gerados para os operadores devido a atuação da proteção.

Em todas as formas de modelagem da proteção, a ação da proteção é simulada, porém, mudanças de dinâmica rápida que surgem durante distúrbios geralmente não são replicadas.

- **Algoritmo de fluxo de potencia:** O algoritmo de fluxo de potência é responsável por calcular os valores de magnitude e ângulo de tensão, bem como fluxos de potencia ativa e reativa no sistema. O algoritmo deve ser executado em um intervalo de tempo semelhante ao tempo de varredura do sistema SCADA. Antes de cada simulação, o PSM deve ser informado se ocorreu alguma alteração nos valores da carga ou da geração, status de equipamentos lógicos e posição de tap de transformadores. Quando isso ocorrer, a configuração do sistema será alterada e utilizada pelo algoritmo de fluxo de potência para que os novos valores sejam atualizados ao módulo CCM.

2.2.2. ES – Educational System

Este módulo é responsável por fornecer ao instrutor métodos e ferramentas para construir e controlar o treinamento, bem como monitorar as ações dos operadores [14]. Através dele o instrutor carrega o estado operativo inicial da rede, tal como: estados de chaves e disjuntores, carregamento do sistema, posição de tap de transformadores, geração das usinas conectadas ao sistema, etc. Esta ferramenta também possibilita que o instrutor selecione uma curva de carga a ser seguida durante o processo de treinamento e que eventos, como por exemplo, abertura de uma linha de transmissão, sejam previamente programados. O ES é utilizado pelo instrutor para acompanhar e controlar a evolução do treinamento, permitindo que o instrutor inicie, pare ou pause o treinamento a qualquer instante. Com este módulo também é possível salvar cenários durante o treinamento, permitindo assim que uma análise posterior seja realizada ou até mesmo submetida em um outro processo de treinamento.

Algumas implementações de OTS utilizam como estado inicial da simulação, o resultado de um estimador de estados, aumentando a confiabilidade e a qualidade dos resultados da simulação.

2.2.3. CCM – Control Center Model

Este módulo é responsável tanto pela réplica do ambiente físico da sala de controle, quanto pela réplica do conjunto de aplicativos que constituem o EMS e SCADA utilizado pelo centro.

Com relação à réplica da sala de controle, podem ser disponibilizadas mesas, cadeiras, computadores, telefones, vídeo walls e tudo mais que for necessário para que o ambiente de treinamento seja o mais semelhante possível ao centro de controle da empresa. Devido ao custo de implementação de uma réplica fiel da sala de controle, muitas empresas optam por uma sala de treinamento mais simples, disponibilizando apenas um pequeno conjunto de computadores e com o auxílio de um projetor para que o instrutor conduza o treinamento. Em alguns casos o treinamento pode ocorrer no próprio centro de controle, alocando-se apenas um subconjunto dos recursos físicos do mesmo para se elaborar o treinamento de um número reduzido de operadores.

Além da infra-estrutura física em que será elaborado o treinamento, o módulo CCM é responsável por disponibilizar ao operador as mesmas funcionalidades das aplicações utilizadas pelo centro de controle para operar o sistema real, ou seja, as aplicações que constituem o EMS e o SCADA da empresa. Dentre as aplicações de EMS, pode-se citar o Fluxo de Potência Ótimo, o Despacho Econômico, o Controle Automático da Geração – CAG, o processador de alarmes, o Sequenciador de Eventos, e outras aplicações de controle de segurança e controle energético. Outro sistema a ser replicado dentro do módulo CCM é o sistema SCADA. O sistema SCADA é o elo entre o operador do centro de controle e o sistema elétrico, pois disponibiliza todos os dados coletados pelo sistema de telemetria e Unidades Terminais Remotas (UTRs) do sistema de automação e possibilita o comando em dispositivos de campo de forma remota. A correta modelagem desse sistema no módulo CCM é essencial para que a réplica do centro de controle seja fiel ao ambiente real de operação, possibilitando realizar tarefas como:

- Abertura e fechamento de chaves e disjuntores;
- Controlar potências reativas e tensões;
- Parametrizar e supervisionar alarmes;
- Gerar a base de dados histórica dos pontos supervisionados;
- Supervisionar o status da comunicação com as diversas UTRs do sistema;

A réplica do sistema SCADA no módulo CCM é responsável por coordenar a comunicação com o módulo PSM, pois as informações que no sistema real viriam das UTRs, no OTS elas são provenientes do módulo de simulação do PSM.

2.3. Fluxo de Informações entre os Módulos do OTS

A Figura 1 ilustra os módulos que compõem o OTS e o fluxo de informação entre eles.

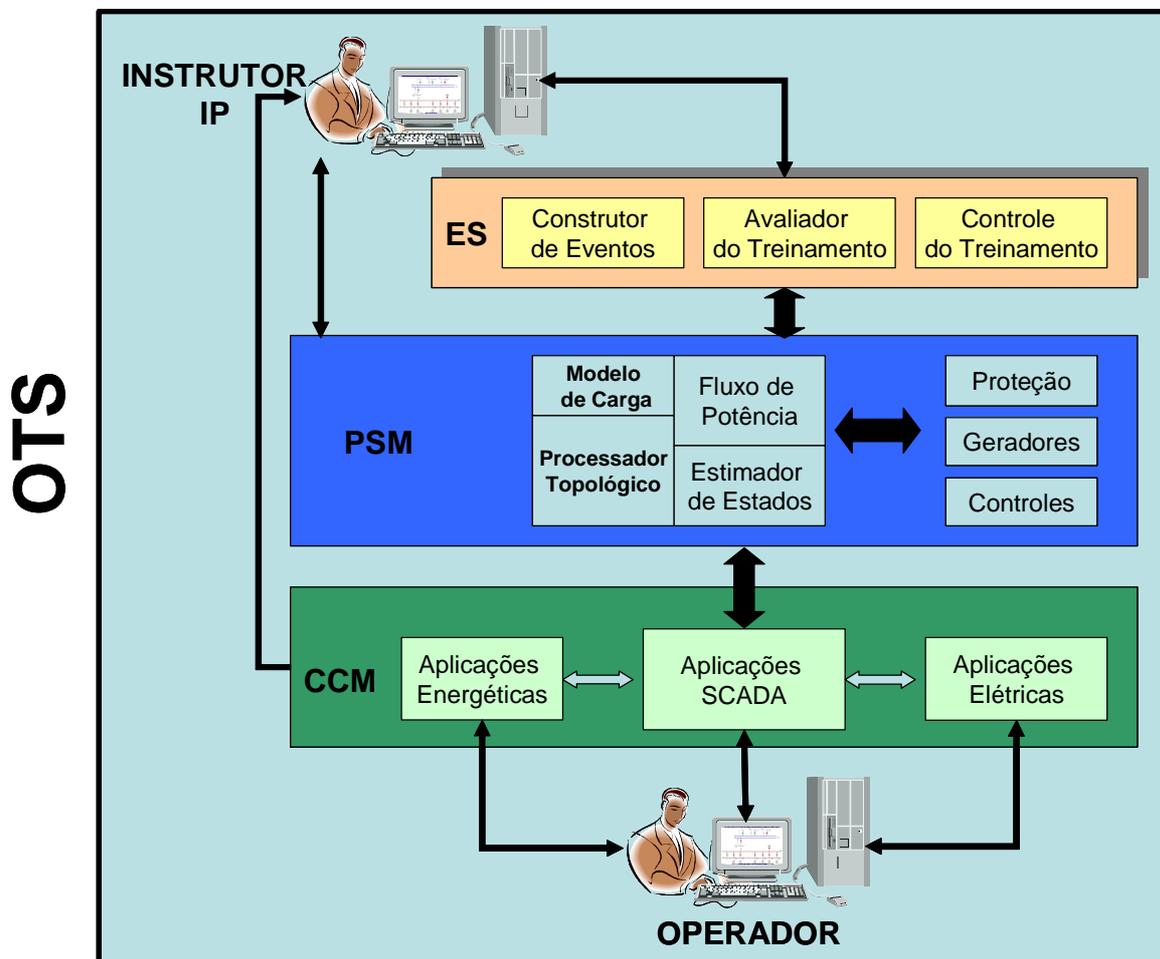


Figura 1: Fluxo de informações de um OTS.

A seguir é apresentado um exemplo de treinamento utilizando os recursos de OTS, bem como principais passos envolvidos na implementação de tal tarefa.

1 - Criação do Cenário Inicial: O instrutor utiliza a posição IP (*Instructor Position*) para criar o cenário inicial e as seqüências de eventos que irão ocorrer a cada instante do treinamento. Para este exemplo é considerada uma curva de geração típica, uma curva de carregamento de um dia de semana de verão e sem nenhuma sobre-carga nas linhas de transmissão. Um evento de abertura de uma linha de transmissão é configurado para ocorrer durante o treinamento.

2 - ES Inicializa a Simulação: Ao iniciar a simulação via ES, os primeiros valores das curvas de carregamento e geração, bem como o estado lógico de chaves, disjuntores e posição de taps de transformadores são enviados ao PSM para que o primeiro ciclo de simulação seja realizado. Assim que terminar o primeiro ciclo, os valores obtidos são informados ao CCM para que o operador possa visualizar os resultados da simulação.

3 - ES Controla a Simulação: Para cada valor instantâneo proveniente das curvas de geração e carregamento, o ES informa o cenário para o PSM para que um novo conjunto de valores sejam simulados e informados ao CCM.

4 - Ocorrência de Evento: No ciclo de simulação em que deve ocorrer o evento de abertura dos disjuntores das extremidades da linha de transmissão, o ES informa ao processador topológico do PSM que esses dois disjuntores abriram. Dessa forma, o PSM pode reconfigurar o sistema e recalcular os valores de potência e tensão que serão apresentados no CCM.

5 - CCM Detecta a Abertura do Disjuntor: Assim que o CCM recebe o novo conjunto de valores calculados pelo PSM após a abertura dos dois disjuntores, os valores analógicos (MW, kV, A) e lógicos (estado dos disjuntores) são automaticamente apresentados aos operadores. Nesse momento, alarmes são acionados pelo processador de alarmes para informar a abertura dos disjuntores e a violação dos limites operacionais das variáveis analógicas.

6 - Operador Toma a Ação de Controle: Considerando que o evento de abertura de linha tenha ocasionado a sobrecarga de uma segunda linha, o operador irá operar outros disjuntores para que a carga seja devidamente distribuída.

7 – CCM Transmite o Comando: Ao detectar que o operador realizou um comando, o CCM envia um sinal para o PSM, indicando que a configuração da rede foi alterada. O PSM calcula o novo conjunto de valores de tensões e potências e informa ao CCM para que o operador possa analisar o resultado de sua ação sobre o sistema.

8 – O Instrutor Avalia a Ação de Controle: Depois do operador sob treinamento ter realizado a ação de controle, o instrutor pode paralisar a simulação para discutir com os outros operadores a ação tomada e as possíveis alternativas disponíveis para controlar o sistema diante do cenário imposto.

9 – O Instrutor Salva o Cenário: Caso seja necessário, o instrutor pode salvar o cenário imediatamente após o evento para que o mesmo possa ser submetido novamente em outra seção ou ser re-avaliado após o treinamento.

2.4. Tipos de Simuladores de Treinamento

A função principal de um OTS é prover condições de treinamento de novos operadores e reciclar operadores em novos procedimentos operacionais. A área de operação de tempo real das empresas é um setor cujos profissionais estão sujeitos a um grande fluxo de informações e o desenvolvimento de suas habilidades para tomar decisões em condições adversas, como por exemplo, quando o sistema encontra-se em condição de emergência, é fator fundamental para restabelecer o serviço e manter um padrão adequado de segurança. Dessa forma, é importante que eventos críticos que podem conduzir o sistema a apresentar riscos de operação, ou até mesmo conduzir a um blecaute, sejam avaliados e simulados antecipadamente. A construção de treinamentos considerando tais cenários é essencial para familiarizar o operador com relação ao comportamento elétrico do sistema quando submetido a diferentes condições [16].

O OTS pode ser configurado de diferentes maneiras para atender as necessidades de treinamento da empresa, podendo este variar devido a aspectos de disponibilidade de *hardware*, limitação de área física, estratégia e pedagogia do treinamento. Uma política eficiente de treinamento deve atender a formação e o aprendizado dos operadores em diversos níveis, desde o treinamento para operadores iniciantes, onde o operador adquire experiência da operação utilizando ações básicas, do tipo abrir e fechar disjuntores e mudar tap de transformadores,

avaliando a resposta estática de suas ações, até operadores e despachantes experientes, na qual o treinamento envolve avaliar respostas estáticas e dinâmicas de um sistema elétrico quando submetido a eventos que o conduzam a um estado de emergência, sendo necessárias ações de controle e proteção para recompor o sistema de forma fluente e coordenada. Adicionalmente, o OTS pode ser utilizado para simular procedimentos operacionais e a resposta do sistema, estática e dinâmica, quando da entrada de subestações, linhas de transmissão ou equipamentos.

Dessa forma, pode-se configurar o OTS de duas maneiras [6]. A seguir é apresentada uma descrição destas configurações de treinamento.

2.4.1. Stand-Alone

Nessa forma de implementação o OTS é um sistema totalmente separado, havendo pouca ou nenhuma ligação com as aplicações que constituem o EMS do centro de controle. Geralmente apresenta um modelo simplificado do centro de controle, podendo o software e *hardware* ser totalmente diferente dos utilizados no sistema real. As interfaces gráficas possuem pouca similaridade com as utilizadas no centro de controle.

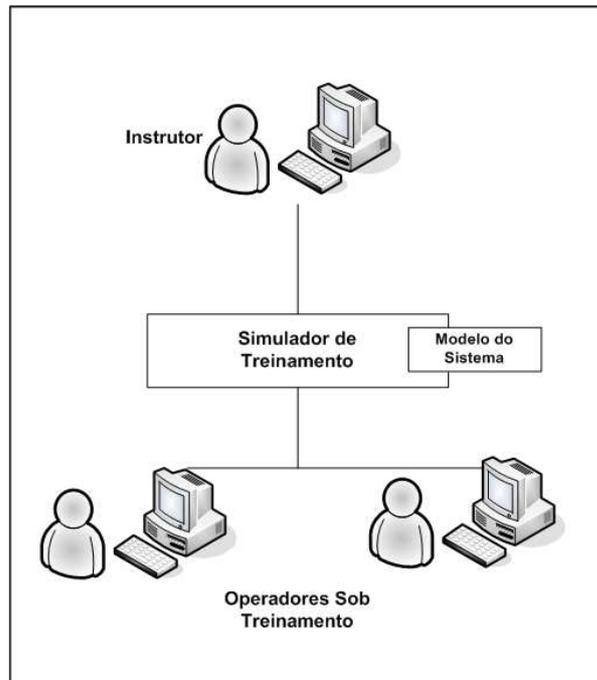


Figura 2: Ambiente OTS na forma Stand-Alone.

Essa forma de implementação de OTS pode tanto ser utilizada por operadores inexperientes previamente instruídos, quanto por operadores experientes que desejam avaliar seus conhecimentos no ambiente de simulação. Também pode ser utilizado pela equipe de planejamento para avaliar e validar procedimentos operativos. Algumas características desse sistema são:

- Livre escolha do fornecedor de OTS no mercado, pois nessa forma de implementação não é necessária a ligação com as aplicações utilizadas no centro de controle;
- Flexibilidade de instalação, pois pode ser instalado em máquinas diferentes das utilizadas pelo centro de controle;
- O mesmo simulador pode ser utilizado por diferentes empresas;
- Geralmente necessita de adaptações em nível de interface gráfica, tratamento de alarmes, mensagens de eventos, etc.;
- Apresenta pouco grau de realismo.

2.4.2. Integrado as Aplicações do Centro de Controle

Nessa forma de implementação o OTS está conectado ou integrado aos sistemas utilizados pelo centro de controle, podendo até mesmo utilizar o mesmo *hardware* do centro de controle (Ex.: O OTS pode ser executado no *hardware* redundante do próprio centro de controle). Nessa forma de implementação o OTS se integra tanto as aplicações (SCADA, Interface gráfica, Fluxo de potência, funções de planejamento, etc.) quanto dados do centro de controle (descrição do sistema elétrico, gráficos, relatórios, medidas elétricas atuais e históricas, etc.). Como consequência dessa integração, algumas características podem ser ressaltadas:

- Funções utilizadas no treinamento são as mesmas utilizadas pelo centro de controle;
- Evita a duplicação de dados, pois gráficos, relatórios e cadastro do sistema elétrico são provenientes do próprio sistema do centro de controle;
- Possibilita que o treinamento seja executado no próprio centro de controle, utilizando parte da infra-estrutura de *hardware* e de software existente;
- O OTS pode ser utilizado para testar novas aplicações antes de serem integradas ao EMS da empresa;
- Pode utilizar informações de tempo real para realizar simulações;

- Dependente do sistema vigente no centro de controle, podendo existir problemas de atualizações e customização as necessidades da empresa.

Embora a função básica do OTS seja o treinamento de operadores e despachantes, ele também pode ser utilizado para estudos de planejamento, possibilitando avaliar o impacto na operação de entrada de novas obras e equipamentos (subestações, linhas de transmissão, banco de capacitores, etc.), avaliar procedimentos operacionais, estudar a saída de operação de equipamentos e testar novas funcionalidades e interfaces.

3. OPC - Object Linking and Embedding for Process Control

As tecnologias relacionadas ao controle e automação industrial estão em constante evolução. Demandas por sistemas flexíveis, reutilizáveis, com velocidade de produção e com baixo índice de falhas são crescentes, bem como as pressões por redução dos custos nos processos industriais. Diante desse contexto, softwares capazes de controlar processos são cada vez mais considerados como um elemento essencial para alcançar tais objetivos [28]. Ao mesmo tempo, mudanças na forma de implementação da automação de chão de fábrica e a disseminação de tecnologias como a Internet e a necessidade da utilização de padrões abertos vem propiciando a inserção de PCs (*Personal Computers*) como uma alternativa viável para construção de sistemas de automação. Atualmente os PCs estão sendo utilizados como coletores de dados, visualizadores de variáveis e controladores de processo. Isso está possibilitando a complementação e/ou a substituição dos tradicionais CLPs (Controlador Lógico Programável) e de terminais de operação. Os motivos desse processo estão relacionados à queda do preço devido à produção em massa de PCs, a permanente multiplicação da capacidade das CPUs (*Central Process Unit*), a disponibilidade de componentes de software mais eficientes e desenvolvidos sob conceitos de usabilidade, bem como a facilidade da integração aos demais sistemas corporativos [30].

A utilização de componentes de softwares flexíveis, capazes de serem reutilizados em ambientes de automação distribuída, também pode ser considerado como um fator importante na busca por redução dos custos envolvidos na implementação, manutenção e integração de tais sistemas [34]. A integração horizontal proporcionada pela comunicação entre dispositivos inteligentes de campo, disponibiliza uma grande quantidade de dados que muitas vezes não são utilizadas por sistemas de mais alto nível devido à falta de integração entre processos de forma horizontal [32]. A falta de tratamento e integração de tais sistemas vem sendo motivo de intensos estudos, uma vez que a integração vertical pode proporcionar poderosas ferramentas de gerência,

unificando dados de processos de fabricação, vendas, desenvolvimento, planejamento e qualidade de produto, possibilitando a integração de sistemas do tipo *Enterprise Resource Planning* (ERP) e *Manufacturing Execution Systems* (MES), como no apresentado em [35]. Porém, para que diferentes fornecedores possam desenvolver softwares capazes de serem conectados via rede a um sistema maior e heterogêneo é imprescindível a existência de padrões de interfaces de comunicação. O *OLE for Process Control* (OPC) é fruto dessa necessidade e atualmente um dos padrões mais adotados tanto por desenvolvedores quanto por usuários [25].

O OPC é um protocolo de comunicação baseado em uma série de padronizações OLE/DCOM (*Object Linking and Embedding / Distributed Component Object Model*), criado e mantido por um consórcio de empresas fabricantes de software e *hardware* de automação industrial, com a finalidade de facilitar a integração entre as diferentes fontes de dados comuns em plantas industriais altamente automatizadas. O OPC é o primeiro componente padronizado desenvolvido para atender as necessidades específicas do domínio da automação [27], padronizando os mecanismos para comunicar com inúmeras fontes de dados disponibilizados por equipamentos de campo e bancos de dados em centros de controle. Dessa forma, os fabricantes de equipamentos de automação podem concentrar esforços no desempenho da comunicação com suas fontes de dados locais e padronizar o mecanismo de acesso de sistemas externos a esses dados. As interfaces OPC disponibilizadas pelos servidores permitem que qualquer cliente acesse seus dados internos de forma padronizada [30].

O desempenho do OPC é considerado adequado para a maioria das aplicações dedicadas e distribuídas que são executadas sobre plataformas computacionais comerciais, como apresentado em [31], onde um servidor OPC foi acessado por quatro clientes, disponibilizando 20.000 valores por segundo. Durante esse ensaio houve um consumo de CPU não superior a 10% de sua capacidade nominal.

O OPC é dividido em diferentes especificações, as quais foram desenvolvidas para atender as necessidades específicas de diferentes problemas relacionados à integração de informações nas indústrias. Tais especificações utilizam um conjunto de funções básicas, que generalizam as necessidades comuns das diferentes especificações, permitindo que as mesmas sejam utilizadas em uma mesma aplicação. As especificações existentes no OPC e suas inter-relações estão apresentadas na figura 3.

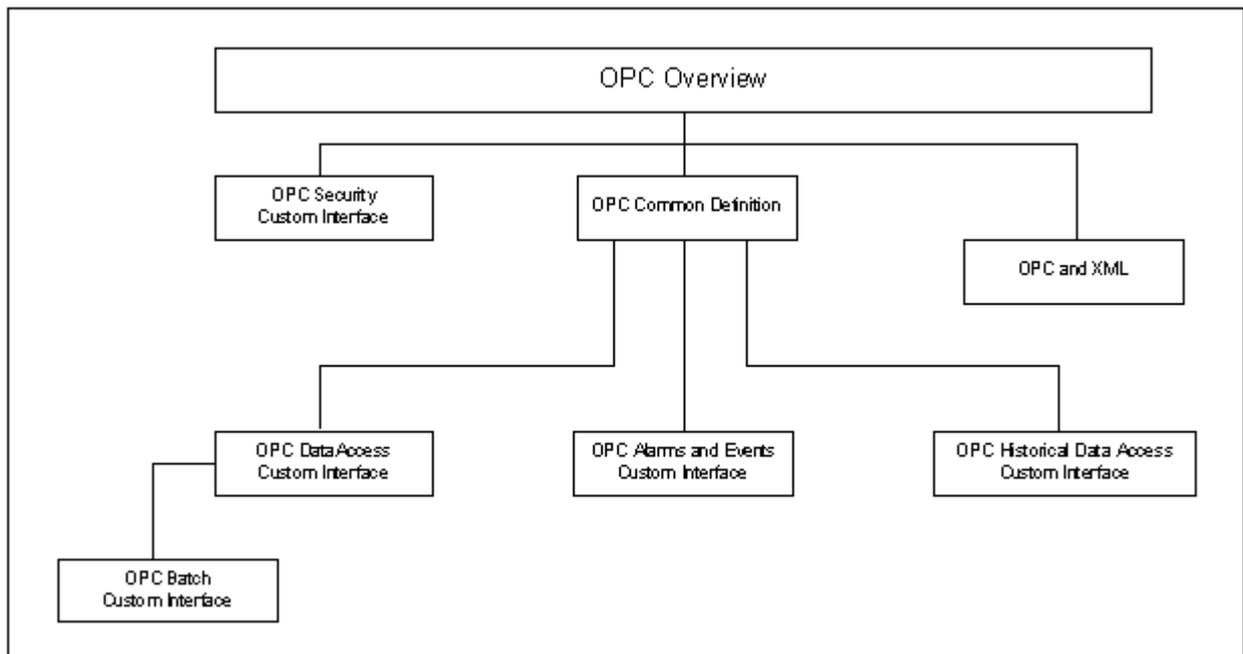


Figura 3: Especificações OPC e suas Inter-relações.

As características, particularidades e aplicação de cada especificação são apresentadas nas seções abaixo. Porém como o OPC foi desenvolvido sobre a tecnologia DCOM, é essencial uma breve explicação sobre o funcionamento desse componente.

3.1. DCOM

O OPC é resultado de uma força tarefa criada para desenvolver um padrão de acesso a dados de tempo real em sistemas operacionais Windows, utilizando a tecnologia desenvolvida pela Microsoft chamada de OLE/DCOM [30].

O *Distributed Component Object Model* é uma tecnologia desenvolvida para possibilitar a comunicação entre componentes de software através de uma rede de comunicação de computadores [29]. Essa tecnologia permite a construção de aplicativos do tipo Cliente-Servidor, onde um mesmo cliente pode acessar diversos servidores e servidores podem ser acessados por diversos clientes ao mesmo tempo [28].

O DCOM está baseado no conceito de interfaces, as quais são responsáveis por exteriorizar os serviços disponíveis em cada objeto através de um grupo de funções definidas. A interface descreve a sintaxe e a semântica dos serviços oferecidos.

A semântica descreve, por exemplo, o correto significado dos argumentos de entrada das funções e os valores admissíveis de retorno. A semântica é normalmente especificada por uma documentação, como por exemplo, o *OPC Specification* [27]. Já a definição da sintaxe é elaborada em uma linguagem especial chamada de IDL (*Interface Definition Language*) [29], a qual define como deve ser a representação binária da interface. Isso possibilita que clientes possam chamar métodos em servidores DCOM independentemente do ambiente em que foram programados. A garantia de compatibilidade binária de objetos DCOM é um dos princípios básicos da tecnologia desenvolvida pela Microsoft.

Objetos DCOM, também chamados de componentes, são instanciados dentro dos servidores. Existem basicamente dois tipos de servidores: os In-ProcServers (DLLs) e os OutProc-Servers (EXEs) [29]. A identificação dos objetos DCOM se dá através do CLSID (*Class Identifier*), sendo este identificador utilizado pelo cliente para acessar o componente DCOM no servidor. Para gerar uma nova instância de um componente no servidor são utilizadas funções disponibilizadas pela biblioteca DCOM. Depois de instanciado um novo componente, o cliente pode requisitar suas interfaces e consecutivamente chamar seus métodos. A localização desse objeto é transparente para o cliente, uma vez que a biblioteca DCOM é responsável por administrar essa tarefa. As informações complementares para a conclusão da tarefa de criação de um componente DCOM são encontradas na *register* do sistema operacional.

O ciclo de vida de um objeto DCOM é controlado por um contador interno. Esse contador é incrementado para cada nova interface requisitada, e decrementado para cada interface liberada pelo cliente. Quando esse contador é zerado, o objeto DCOM é destruído automaticamente.

3.2. OPC Overview/OPC Common Definition and Interfaces

Durante a especificação do OPC, percebeu-se que existiam definições e necessidades comuns aos diferentes campos de aplicação a serem abrangidos pelo padrão. Tais definições geraram as especificações *OPC Overview* e *OPC Common Definitions and Interfaces* [27], as quais definem os modelos de objetos, as interfaces, os métodos, os parâmetros e os conceitos para os diferentes campos de aplicação.

O *OPC Overview* tem a função de servir como uma documentação informativa a respeito dos diferentes campos da aplicação, suas tecnologias básicas e as versões das especificações disponíveis. A especificação *OPC Common Difinitions and Interfaces* trata de aspectos de instalação e registro, identificação de servidores OPC em uma rede de computadores, adaptação de interfaces para diferentes linguagens de programação que não suportam ponteiros de função e regras para a transferência de eventos de finalização de processo entre o Servidor e o Cliente OPC.

3.3. OPC Data Access

O *OPC Data Access* (OPC DA) [40] foi a primeira especificação OPC a ser concluída. Seu objetivo é definir uma interface de acesso, entre clientes e servidores, capaz de disponibilizar dados de processo. Servidores OPC DA permitem que diversos clientes OPC DA acessem, de forma transparente, valores de variáveis internas (Ex.: temperatura de uma caldeira) e executem escritas em valores de controle (Ex.: comando de abertura de uma válvula). Tais valores passíveis de leitura e escrita podem ser provenientes de cartões de Entrada e Saída (do inglês *I/O cards*) conectados a um PC, ou até mesmo de PCs conectados a redes industriais de comunicação de dados como: Profibus, DNP3, IEC60870.5, MODBUS, etc.

A especificação OPC DA introduz o conceito de *Namespace* e hierarquia de objetos, que podem ser implementados no servidor para que o cliente extraia informações contextualizadas, e não apenas valores de variáveis. O *namespace* pode ser implementado de maneira hierárquica, na forma de árvore, ou de forma plana. Na forma hierárquica, os nós podem ser utilizados no sentido de organizar os dados disponíveis no servidor, possibilitando que, por exemplo, a estrutura de árvore represente os dispositivos em um sistema de coleta de dados. Já as folhas, também chamadas de itens, as quais são disponibilizadas pelos nós, representam os repositórios de dados de leitura e escrita. Atributos podem ser implementados nos nós e folhas, sendo possível, por exemplo, especificar o nome do fabricante de um determinado dispositivo.

Um *Data Access Client* pode criar diversos objetos OPC em um *Data Access Server*. A estrutura hierárquica dos objetos OPC no servidor se dá em três níveis: o *OPCServer Object* no primeiro nível, o *OPCGroup Object* no segundo nível e no último nível o *OPCItem*. Os *OPCItems* representam os nós e folhas do *namespace*, já os *OPCGroups* são utilizados para estruturar os *OPCItems*. Essa estruturação em grupos pode ser utilizada seguindo critérios que

variam dependendo do tipo de aplicação. Tais critérios podem, por exemplo, estarem relacionados a aspectos lógicos (todas as variáveis de um determinado setor de produção) ou aspectos dinâmicos (variáveis de processo com comportamento temporal similar) ou a algum critério específico do processo monitorado.

OPCGroup Objects são gerenciados por *OPCServer Objects*, podendo haver diversos grupos instanciados em um mesmo servidor, sendo esse numero dependente do número de clientes conectados. Para cada cliente conectado, um ou mais grupos são instanciados no servidor, obedecendo a estrutura requisitada pelo *OPC Client*.

O número de níveis hierárquicos para objetos OPC em um servidor é fixo e sua montagem é dependente do cliente, porém os níveis hierárquicos do *Namespace* dependem da implementação do servidor, e será a mesma para todos os clientes conectados. A figura 4 mostra uma hierarquia de objetos e as suas relações com os itens do *Namespace*. No exemplo apresentado, o cliente OPC instanciou dois *OPCGroup Objects* e para cada *OPC Group*, dois *OPCItem Objects*.

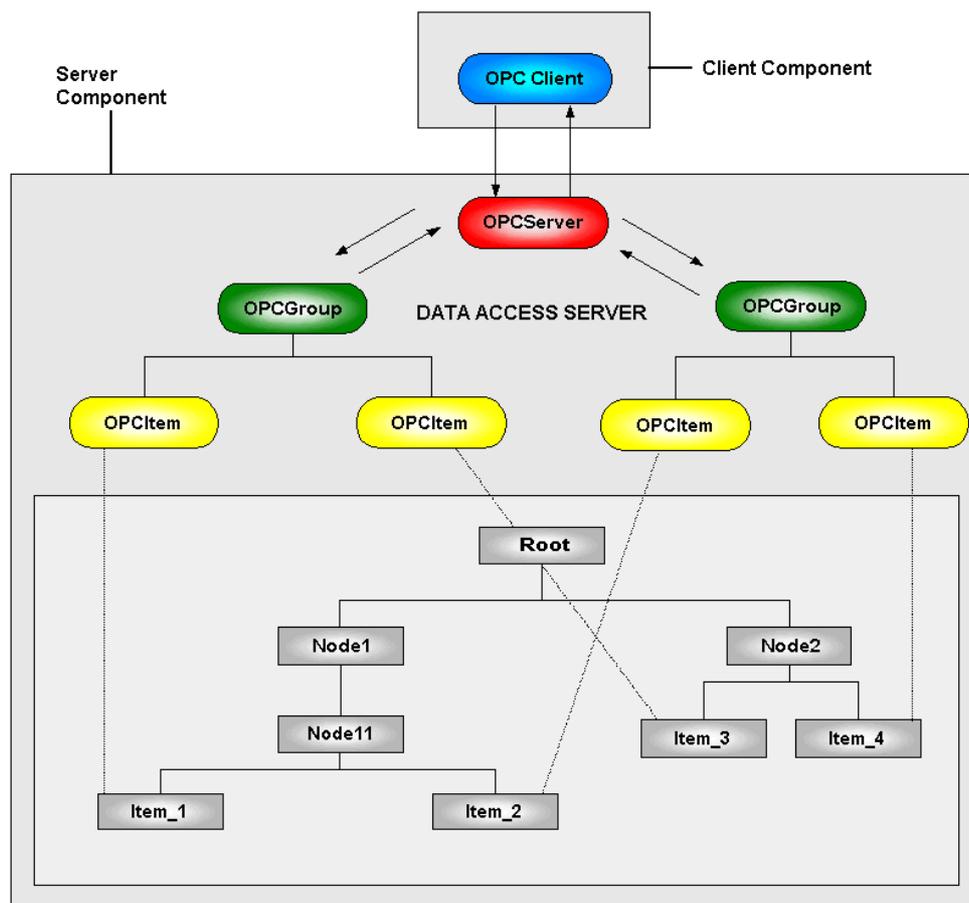


Figura 4: *Namespace* e Hierarquia de Objetos em um Servidor Data Access.

3.4. OPC Alarms and Events

A especificação *Alarms and Events* [41] define uma interface cliente/servidor que possibilita a monitoração e reconhecimento de eventos e alarmes. O *Alarms and Events* podem gravar e avaliar valores de diferentes fontes de dados, identificando as ocorrências de eventos. As fontes de dados podem estar localizadas em cartões de Entrada e Saída, conectados a PCs, ou em um dispositivo como um CLP (Controlador Lógico Programável) conectado por um canal serial ou uma rede industrial.

Apesar de o OPC Alarms and Events utilizar as mesmas fontes de dados que o *OPC Data Access*, algumas características fazem com que sejam estabelecidas duas especificações diferentes. A primeira característica importante é que servidores *Data Access* disponibilizam um fluxo de dados quase que constante, onde a transferência automática de dados obedece a uma banda morta previamente especificada. A segunda está no fato de que servidores *Data Access* não mandam valores para clientes, mas sim informação de que algo anormal ocorreu, como por exemplo, que uma temperatura violou um limite estabelecido e que deve ser tratada como uma situação de alarme.

Servidores *OPC Alarms and Events* podem ser implementados como componentes independentes, que podem obter informações diretamente de aplicações e de dispositivos de campo, ou até mesmo de servidores *OPC Data Access*. Nessa última situação, torna-se necessária a implementação de um cliente *OPC Data Access* junto ao servidor *OPC Alarms and Events*.

A especificação *Alarms and Events* suporta o fato de que um cliente pode acessar mais de um servidor e que um servidor possa ser acessado por mais de um cliente ao mesmo tempo.

Um evento é definido pela especificação como sendo a detecção de ocorrências como falha de equipamentos, intervenções de operadores e violação de limites pré-estabelecidos, onde tais limites são estabelecidos por quatro patamares: Emergencial Superior (HiHi), Operacional Superior (Hi), Operacional Inferior (Lo) e Emergencial inferior (LoLo). O evento ocorre se uma sub-condição torna-se ativa, como por exemplo, uma temperatura violou um determinado limite. Nesse instante, o evento sinaliza a ocorrência de uma situação de alarme, a qual é sustentada enquanto a subcondição estiver ativada.

3.5. OPC Historical Data Access

Servidores *OPC Historical Data Access* [42] permitem que clientes OPC acessem valores históricos armazenados em suas bases locais.

Esta especificação permite que dois tipos de servidores sejam implementados:

- Servidores de tendência: suporta apenas algumas interfaces para acesso aos dados;
- Servidores de análise e compressão de dados: Permite realizar análise de valores como, por exemplo, valor médio, valor mínimo e valor máximo. Também é permitida a regeneração de valores e adição de informações complementares.

O *Historical Data Access Specification* define duas estruturas que podem ser implementadas pelo servidor para serem utilizadas pelos clientes: o *namespace* e a hierarquia de objetos.

O *namespace* contém todos os dados que podem ser acessados no servidor e sua semântica é similar a encontrada no *Data Access Server*.

Um cliente *OPC Historical Data Access* pode criar diferentes objetos OPC no servidor. O *OPCHDAServer Object* é o objeto de mais alto nível na estrutura hierárquica de objetos, sendo através dele, disponibilizadas as funcionalidades de leitura, escrita e alteração de dados. Já as funcionalidades de localização de *namespaces* é oferecida através do *OPCHDABrowser Object*.

Diferentemente do *Data Access Server*, o *Historical Data Access* não implementa o conceito de *OPCGroup* e *OPCItem*, permitindo que os clientes manipulem seus dados internos sem a necessidade de criação de objetos. Enquanto o *Data Access* possibilita o acesso permanente às variáveis de processo, criação e remoção de grupos e o acesso ao mesmo *namespace* por mais de um Item, o *Historical Data Access* permite que os clientes acessem diretamente o conjunto de valores requisitados, pois os clientes raramente estão interessados em um único valor. Geralmente, consultas ao histórico são realizadas em uma frequência menor que as necessárias nos processos de tempo real e retornam um volume de dados consideravelmente maior, ou um agregado, como por exemplo, um valor médio.

A figura 5 mostra a hierarquia de objetos e sua relação com o *namespace*.

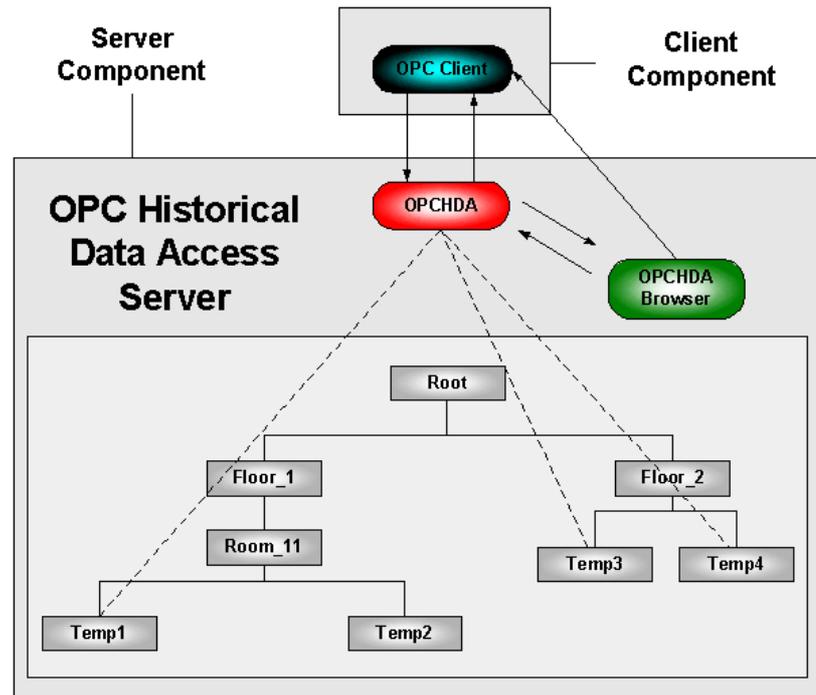


Figura 5: Namespace e Hierarquia de Objetos no Historical Data Server.

3.6. OPC XML-DA

O grupo de trabalho responsável pela especificação do OPC XML [43], criado em março de 2000, tem como missão integrar o OPC às tecnologias Microsoft BizTalk e WEB, através do uso do XML. O eXtensible Markup Language (XML) [44] é uma linguagem descritiva de dados altamente flexível, de fácil entendimento e aprendizagem. Esta linguagem é considerada a sucessora do HTML (Hypertext Markup Language). O XML é muito utilizado para descrever dados de aplicações específicas, inclusive na área de automação, onde pode ser facilmente encontrada em documentações de gerência de processos. Devido a sua versatilidade e potencialidade, o XML foi classificado pela OPC Foundation como uma importante tecnologia para a comunicação de dados OPC através da Internet.

Os documentos XML são estruturados de forma a conter elementos, os quais possuem atributos e também podem conter outros elementos. O documento XML deve obedecer à sintaxe XML, sendo considerado válido quando obedecer ao modelo especificado no documento chamado de esquema. O esquema estabelece um conjunto de regras que definem a composição do documento a partir de uma gramática formal [45], detalhando quais elementos podem ser

utilizados e onde eles podem ser aplicados. A preparação de documentos e esquemas XML, bem como a validação e processamento de arquivos, é suportado por um grande número de ferramentas disponíveis no mercado.

Um exemplo prático da utilização do XML na indústria pode ser o armazenamento das descrições das características físicas dos equipamentos de uma determinada planta industrial. Através do XML, fabricantes, integradores e usuários definem um esquema que determina a estrutura da descrição dos dados. Dessa forma, o fabricante informa os dados do equipamento através de uma ferramenta existente. O integrador e o usuário final armazenam esses dados em um banco de dados, disponibilizando-os para futuras tarefas como: comissionamento de aplicações, manutenção e operação.

Durante o desenvolvimento do OPC XML-DA, algumas características foram consideradas essências, como:

- A conexão entre clientes e servidores via Internet não deve ser permanente para a maioria dos casos, devendo a mesma ser fechada logo após a requisição dos dados;
- O número de clientes que podem acessar o servidor via Internet pode ser muito maior que o número de clientes que acessam via Intranet;
- Após a desconexão do cliente, todas as informações referentes à conexão estabelecida são totalmente apagadas não restando nenhuma configuração remanescente.

A figura 6 apresenta o modelo de estrutura à qual o OPC XML-DA foi desenvolvido para suportar.

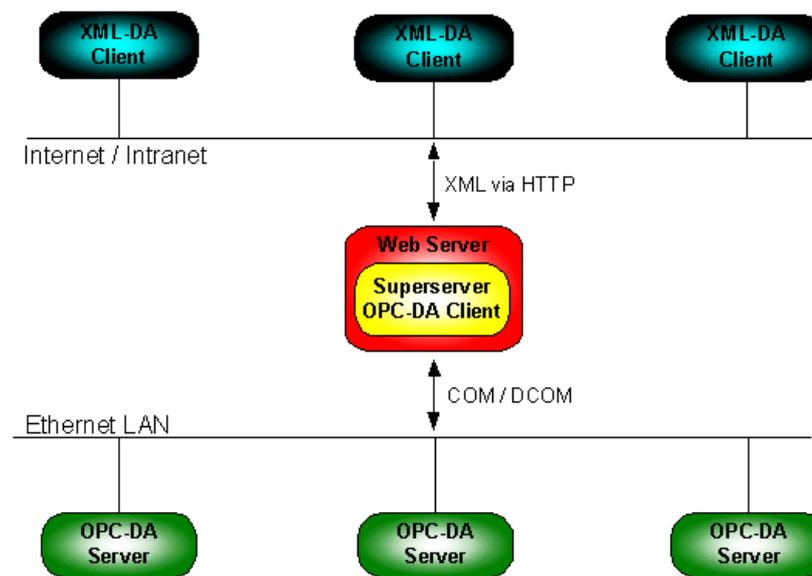


Figura 6: Integração de informações disponibilizadas por servidores OPC-DA para a Internet via OPC XML-DA

O componente servidor web, chamado de *superserver*, é responsável por gerenciar a troca de documentos XML com os clientes web conectados a ele via Internet. O conteúdo do documento é estruturado de forma a conter a chamada apropriada dos métodos disponibilizados nas interfaces dos objetos OPC existentes no servidor. Os resultados dos métodos chamados são retornados aos clientes através de documentos XML. Ao requisitar dados de um *OPC Data Access Server* específico, o documento XML deve conter informações suficiente para instanciar um objeto OPC, adicionar um *OPCGroup* e um *OPCItem*. Assim que os valores requisitados pelo cliente estiverem disponíveis no servidor, um novo documento XML é criado e enviado ao cliente através da Internet. Após o envio do documento, todos os objetos OPC que tinham sido criados pelo cliente são destruídos no servidor.

4. Construção dos Módulos PSM e ES

Neste capítulo é apresentada a ferramenta computacional utilizada para realizar a função do módulo PSM e ES do ambiente OTS proposto neste trabalho. São apresentadas as características do programa utilizado para modelar e simular sistemas elétricos, as adaptações necessárias para a implementação de um servidor OPC-DA responsável por disponibilizar as informações simuladas ao módulo CCM, bem como um sistema exemplo implementado sobre os módulos PSM e ES para avaliação do sistema OTS proposto.

4.1. FASEE: Framework para Análise de Sistemas de Energia Elétrica

O Grupo de Sistemas de Energia Elétrica (GSEE) da PUCRS vem desenvolvendo a alguns anos ferramentas computacionais para gerenciamento, simulação e análise de sistemas elétricos, tanto para operação off-line quanto para operação on-line. Vários destes sistemas estão atualmente implantados em diferentes empresas do setor elétrico (em nível de geração, transmissão e distribuição) com reconhecida aprovação pelos engenheiros do setor. Dentre os sistemas desenvolvidos pelo GSEE, destaca-se a plataforma FASEE como ambiente computacional de desenvolvimento e suporte para a construção das ferramentas computacionais implementadas pelo GSEE.

A plataforma computacional FASEE consiste de um ambiente de desenvolvimento orientado a objetos para simulação e análise de sistemas de energia elétrica. Este ambiente define um conjunto de classes e objetos cooperantes, que implementam funcionalidades comuns a diversos tipos de aplicativos na área de sistemas de energia elétrica, definindo assim uma estrutura computacional geral (Modelo Orientado a Objetos) que pode ser utilizada como base para a construção de um amplo conjunto de ferramentas na área de sistemas elétricos de potência. A construção de um aplicativo específico no ambiente FASEE necessita apenas da customização

das funcionalidades básicas do ambiente FASEE, juntamente com a adição das características particulares do problema em questão. Toda a arquitetura padrão do projeto (sua estrutura geral, divisão em classes e como estas colaboram entre si) são pré-determinadas pelas classes e objetos base do ambiente FASEE. Desta forma, o engenheiro pode concentrar-se nos aspectos específicos da sua aplicação, o que conduz a construção mais rápida e eficiente de aplicativos. Dentre as principais características do ambiente FASEE destacam-se:

- Gerenciamento da descrição topológica da rede elétrica: A descrição topológica define o arranjo estrutural de um sistema (áreas, subestações, etc.), seus dispositivos e equipamentos (geradores, cargas, linhas de transmissão, etc.) bem como os relacionamentos e suas interconexões;
- Gerenciamento da estrutura resultante do processo de configuração da rede elétrica: Nesta descrição os dispositivos lógicos (seccionadoras, disjuntores, etc.) não são representados. O sistema é reduzido a *Barras Elétricas* e dispositivos efetivamente conectados a estas barras, determinando a configuração operativa atual do sistema;
- Utilização de uma estrutura computacional única para todos os aplicativos finais: Facilita o desenvolvimento de ambientes integrados constituídos de diversas ferramentas computacionais em um mesmo ambiente;
- Facilidade para o desenvolvimento, atualização e expansão dos aplicativos: Agiliza a inclusão de novos modelos, equipamentos e metodologias de simulação e análise;
- Equipamentos e modelos definidos pelo usuário, independente de sua complexidade ou topologia: Esta característica permite grande flexibilidade para o desenvolvimento e inclusão de novos modelos ao sistema;
- Novas ferramentas adicionadas ao ambiente passam a ser disponíveis para todos os aplicativos desenvolvidos no ambiente: Uma ferramenta para gerenciamento de Algoritmos Genéticos, por exemplo, incorporada ao ambiente é automaticamente disponibilizada para todos os aplicativos desenvolvidos no ambiente.

Atualmente a plataforma computacional FASEE conta com um amplo conjunto de aplicativos já implementados, tais como:

- Configurador de redes elétricas;

- Fluxo de potência (método de Newton, formulado em coordenadas polares ou retangulares, e método desacoplado rápido);
- Coeficientes de sensibilidade;
- Fluxo de potência generalizado (Full-Newton, formulado em coordenadas polares ou retangulares, e injeção de potência ou corrente na rede elétrica);
- Análise modal (cálculo de autovalores e autovetores do sistema dinâmico, fatores de participação e mode-shapes);
- Simulação Completa para Análise da Estabilidade Transitória (formulado pelos métodos alternado implícito ou simultâneo implícito);
- Simulação Rápida para Estudos de Estabilidade de Longo Prazo (método quase-estático com representação da dinâmica completa do sistema);
- Fluxo de Potência para Redes de Distribuição (método de soma de potências);
- Reconfiguração ótima para Redes de Distribuição (algoritmos genéticos)
- Alocação de Bancos de Capacitores em Redes de Distribuição;
- Recomposição de Redes de Distribuição sob Distúrbio;

4.2. Ferramenta de Modelagem e Construção de Ambientes de Simulação de Sistemas Elétricos

Como o FASEE consiste de um framework para desenvolvimento de aplicativos de análise elétrica, ou seja, não apresenta uma interface gráfica e construção, manipulação e visualização de seus dados internos, foi desenvolvida uma parceria entre a empresa PowerSysLab Engenharia e Sistemas Ltda. e o GSEE para desenvolver um ambiente gráfico de construção de aplicativos elétricos chamado de PSL[®]-SDK. No PSL[®]-SDK foram desenvolvidas ferramentas de modelagem e parametrização da estrutura de dados do FASEE, facilitando a interface entre o projetista e o framework FASEE.

Através do PSL[®]-SDK é possível construir, graficamente, o modelo de um sistema elétrico através de um conjunto de ferramentas que permitem:

- Criar e configurar os equipamentos que constituem um sistema elétrico (disjuntores, capacitores, reatores, linhas de transmissão, transformadores, geradores, reguladores de tensão, etc.);
- Determinar a conexão entre os equipamentos;
- Construir os sinóticos para operação do ambiente de simulação (diagrama das subestações e visualização de conexão entre subestações);
- Incluir medidores para visualização de variáveis elétricas simuladas;
- Implementar rotinas de carregamento de valores históricos (provenientes do SCADA) de variáveis analógicas (como tensão, potencia ativa e reativa) e digitais (estado de chaves e disjuntores) para construção de cenários operativos;
- Simular operações elétricas como abertura/fechamento de chaves e disjuntores, alteração de posição de TAP, inclusão/exclusão de banco de capacitores, atuação de proteções de sobre corrente, etc.

Uma vez que o conjunto de funcionalidades disponibilizadas pelo FASEE e o PSL[®]-SDK suportam grande parte das funcionalidades que o módulo PSM do OTS deve possuir, foi decidido utilizar estes softwares como ferramentas para a implementação do ambiente de simulação. Para isto, a PSL disponibilizou uma versão de estudo, com parte do código fonte aberto, possibilitando implementações de teste e avaliação. Com esta versão foi possível implementar interfaces de dados e telas de operação.

4.3. Módulo ES sobre o PSL[®]-SDK

O PSL[®]-SDK disponibiliza algumas ferramentas de construção de cenários e monitoração da simulação que podem ser utilizadas na construção do módulo ES. A primeira característica, como citado anteriormente, são as ferramentas customizáveis de importação de cenários passados. Essa ferramenta permite que através de consultas elaboradas em SQL, valores históricos de variáveis analógicas e digitais sejam carregados, possibilitando a reconstrução de cenários operativos passados. Para isto, a ferramenta disponibiliza um mapa onde pode ser configurada a relação entre os medidores criados no módulo PSM com os medidores existentes no sistema SCADA.

O módulo ES também disponibiliza uma ferramenta para criar curvas de carga manualmente, de forma a ensaiar diferentes comportamentos temporais. Com essa ferramenta, todas as cargas criadas no módulo PSM seguem um comportamento temporal único, determinado pela curva criada. A figura 7 apresenta uma curva de carga criada para um determinado sistema exemplo.

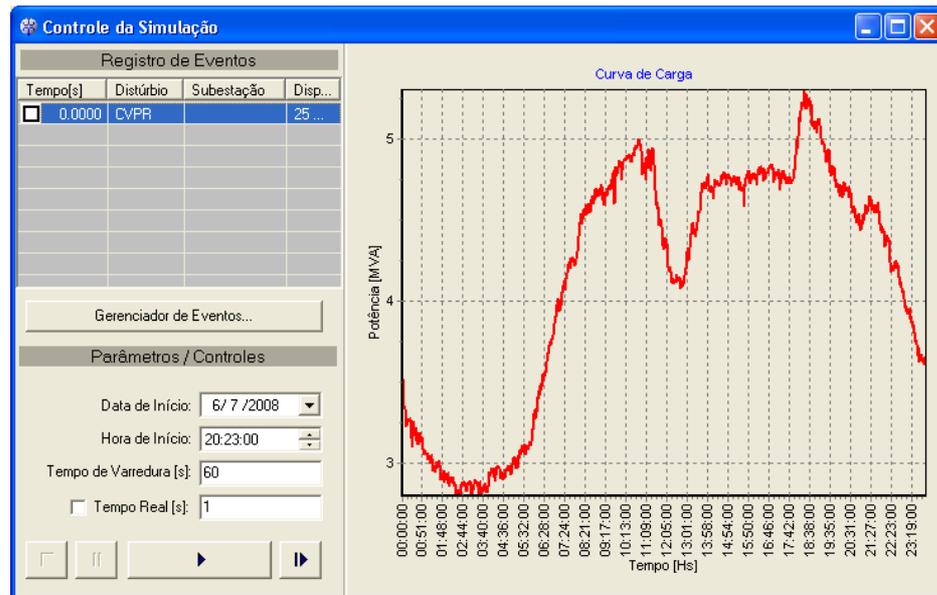


Figura 7: Curva de carga criada através do módulo ES do PSL[®]-SDK.

Também são disponibilizadas ferramentas para criação de eventos de abertura e fechamento de disjuntores, degrau de carga e geração, bem como atuação de proteções.

Todas essas ferramentas são disponibilizadas ao usuário juntamente com um controlador de simulação que permite determinar o período passado a ser simulado (data de início) e a escala de tempo para a simulação. Com estas ferramentas é possível, por exemplo, simular todos os eventos ocorridos durante um mês em poucas horas de simulação.

4.4. Adaptação de um Servidor OPC ao PSL-SDK

Através da utilização dos módulos de desenvolvimento disponibilizados pela versão de estudo do PSL-SDK, foi implementado um servidor OPC-DA acoplado ao módulo PSM. Este

servidor OPC foi desenvolvido para disponibilizar os dados resultantes da simulação para diferentes clientes conectados a mesma rede de computadores a qual o PSM está sendo executado. O servidor OPC também permite que operações de escrita sejam realizadas, possibilitando comandos de controle sobre equipamentos como: disjuntores, chaves, reguladores de tensão, geradores e transformadores. Também através do servidor OPC foram desenvolvidos pontos específicos para controlar a simulação, permitindo que processos remotos gerenciem a simulação no PSM.

Tanto o FASEE quanto o PSL-SDK foram desenvolvidos na plataforma de programação Borland, chamada de C++ Builder 6. Essa plataforma apresenta um ambiente completo de construção, compilação e depuração de códigos desenvolvidos em linguagem de programação orientada a objetos chamada de C++. Outra característica importante dessa plataforma está no fato de serem disponibilizadas bibliotecas que facilitam o desenvolvimento de tarefas específicas. Essas bibliotecas são comercializadas por diversos fornecedores e possibilitam que as mais diversas necessidades, em nível de programação, sejam solucionadas de forma rápida e eficiente. Dessa forma, o desenvolvedor se preocupa apenas em planejar a aplicação final, sem desperdiçar tempo em tarefas que não são o objetivo do projeto.

Diante dessas características e da complexidade envolvida no desenvolvimento de um servidor OPC-DA, foram verificados alguns fornecedores certificados de kit de desenvolvimento OPC para a plataforma Borland® C++ Builder 6 no site oficial da OPC Foundation [25]. Dentre todos os avaliados, o único que apresentou os requisitos necessários para tornar o módulo PSM um servidor OPC-DA foi o Prosys Sentrol™, comercializado pela Prosys Service and Software Solution. Além de possuir todas as características necessárias, o Prosys Sentrol disponibiliza uma versão de avaliação sem custo para estudantes, o que viabilizou a implementação do servidor nesse trabalho.

A implementação do servidor OPC-DA através da utilização do kit de desenvolvimento requisitou apenas conhecimentos básicos sobre o funcionamento do protocolo OPC, do ambiente de programação Builder 6 e da estruturação de dados do FASEE. Para organizar os itens do *namespace* dentro do servidor OPC, foi obedecida uma convenção para facilitar a localização de variáveis via *OPC Browser*. A convenção para implementação do *namespace* seguiu a estrutura apresentada na Figura 8.

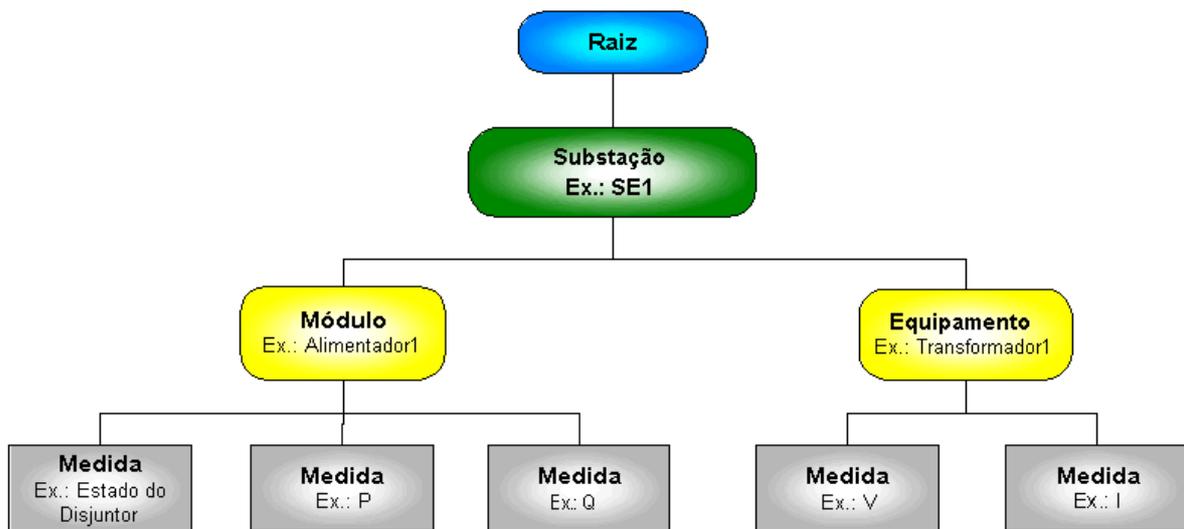


Figura 8: Hierarquia dos dados internos no servidor OPC do PSM.

Na estrutura do *namespace* criada, o primeiro nível abaixo da raiz é constituído de subestações. Logo abaixo das subestações estão os módulos de manobras, que são um conjunto de equipamentos necessários para a conexão de linhas de transmissão, transformadores, equipamentos de compensação de potência reativa ou interligação de barramentos [46]. No mesmo nível dos módulos de manobras, estão os equipamentos elétricos como transformadores, banco de capacitores, reatores, reguladores e geradores. O quarto nível da hierarquia do *namespace* são as medidas pertencentes aos módulos e equipamentos.

O módulo servidor OPC foi desenvolvido de forma a coletar e disponibilizar, em sua interface, todas as medidas simuladas pelo PSM. Toda vez que um novo fluxo de potência é executado pelo PSM (de um em um segundo), as variáveis que sofrem alterações são reportadas para o servidor, o qual se encarrega em atualizar os valores para os clientes OPC conectados. Para elaborar as variáveis responsáveis por receber comandos (Ex.: Pontos de escrita de estado de disjuntores e tap de transformadores), foram criados pontos no servidor capazes de detectarem o evento de variação devido à escrita de um novo valor. Ao detectar o evento de escrita, o comando é interpretado e o conjunto de valores resultantes do fluxo de potência, considerando o novo cenário, são disponibilizados no servidor no ciclo seguinte de simulação.

4.5. Modelagem de um Sistema Exemplo sobre o FASEE

Depois de implementadas as funcionalidades necessárias para que o FASEE/PSL[®]-SDK apresentasse as funções básicas de um PSM e fosse capaz de disponibilizar os dados de simulação ao módulo CCM via OPC-DA, foi decidido modelar um sistema exemplo sobre a plataforma, de modo a avaliar as potencialidades de treinamento que poderiam ser extraídas da estrutura proposta. O sistema exemplo modelado, disponibilizado em [47], possui 28 subestações, 45 barras de tensão, 55 linhas de transmissão, 17 transformadores, 735 elementos de manobra, 8 geradores, 3 banco de capacitores e 2 reatores. No sistema exemplo foram alocados 1258 medidores, obtendo valores de estado de chaves e disjuntores, posição de TAP de transformadores, tensões e potências ao longo da rede.

Cada elemento do sistema foi construído sobre a plataforma de teste, implementado-se a conectividade entre os equipamentos que compõem o sistema elétrico, a parametrização de seus modelos elétricos e a alocação dos medidores no sistema. A figura 9 apresenta a tela de visualização do sistema, representado as 28 subestações e as 55 linhas de transmissão.

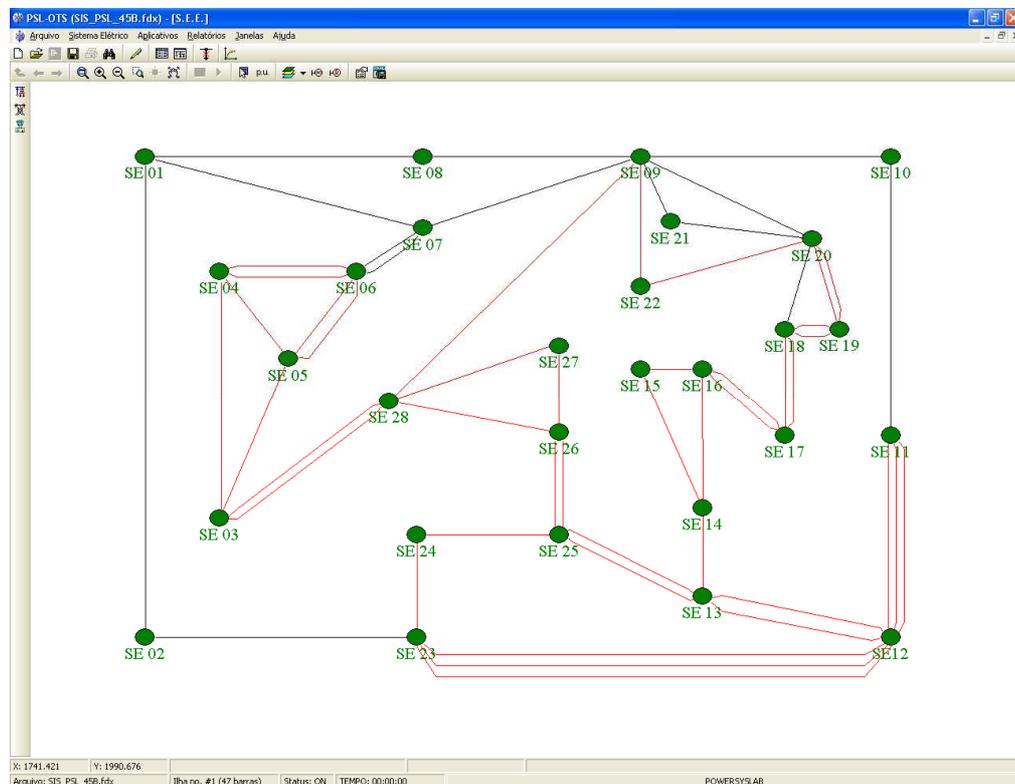


Figura 9: Tela de representação do sistema exemplo implementado no módulo PSM.

A conectividade entre os terminais dos equipamentos que constituem cada subestação foi implementada sobre o módulo PSM, bem como a parametrização dos modelos de cada elemento de rede. Como não é objeto desse trabalho a modelagem dos elementos elétricos, foram utilizados os modelos disponibilizados pelo PSL[®]-SDK, e os mesmos foram apenas parametrizados durante a implementação das subestações. A figura 10 apresenta o diagrama unifilar da subestação 4 do sistema exemplo e a tela de configuração dos parâmetros de conectividade e parametrização do modelo do transformador da subestação.

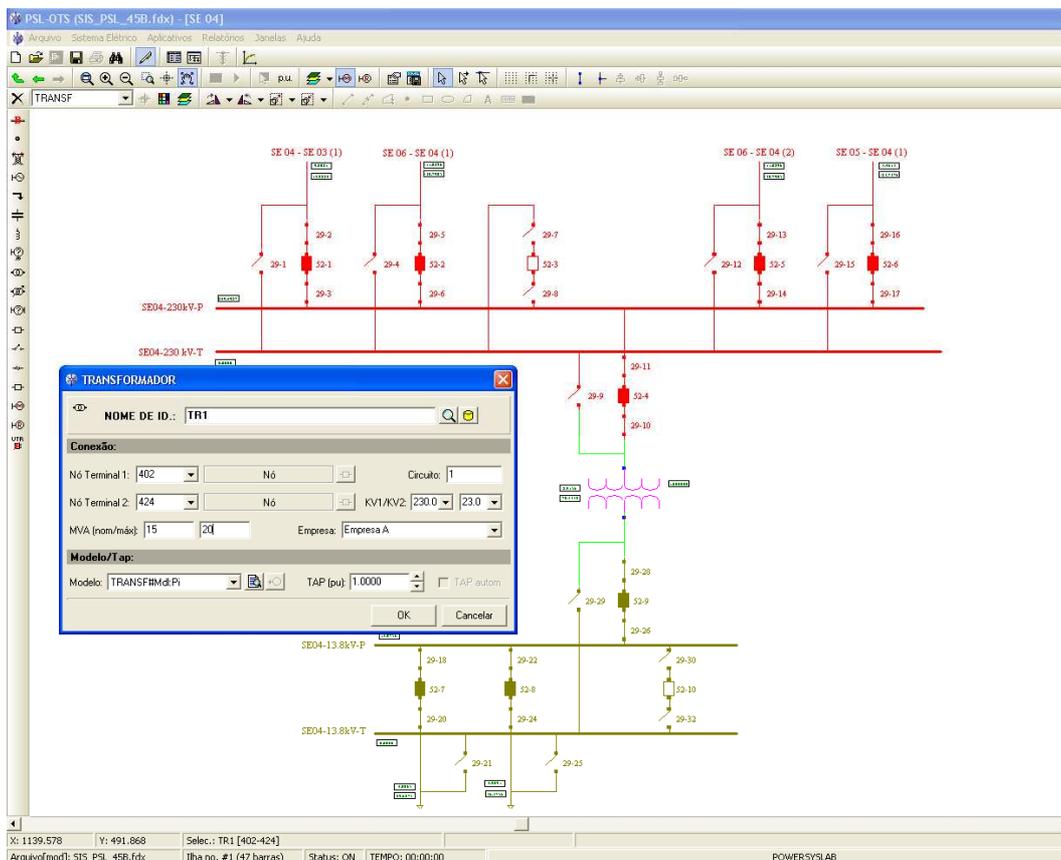


Figura 10: Tela de implementação do modelo da subestação 4 do sistema exemplo.

Depois de modeladas as subestações e as linhas de transmissão do sistema, foi construída uma curva típica de carregamento, que leva em consideração apenas a variação temporal que cada elemento de carga do sistema deveria seguir durante a simulação de um período de 24 horas. A curva foi padronizada para que o valor máximo possuísse valor igual a 1 no horário de ponta. Como em cada elemento de carga alocado no sistema foi configurado apenas um valor de potência ativa e um valor de potência reativa, a curva horária de carga é obtida através da

multiplicação dos valores da curva pelos valores de potências configurados nos elemento de carga.

Com o sistema modelado, foram elaborados testes utilizando as funções de processamento topológico e de fluxo de potências, de forma a avaliar a conectividade entre os elementos, a qualidade dos resultados e a convergência dos algoritmos. Com o sistema comissionado, foram realizados testes para avaliar a simulação seguindo a curva de carregamento construída e a escala de tempo configurada para a simulação.

Com o auxílio da ferramenta de configuração de escala de tempo foi possível simular 24 horas de valores de potência ativa e reativa em apenas 24 minutos, configurando uma escala de 1min:1hora, ou seja, um minuto de simulação varre o comportamento de carga de uma hora. Outra característica ensaiada foi a criação de *scripts* de abertura de linhas de transmissão para um determinado tempo de simulação, possibilitando a simulação da perda de linhas de transmissão durante a simulação, sem a necessidade de intervir manualmente. Na figura 11 é apresentada a tela de configuração de script de abertura do disjuntor 52-1 da subestação 1 as 10:00:00 do dia simulado.

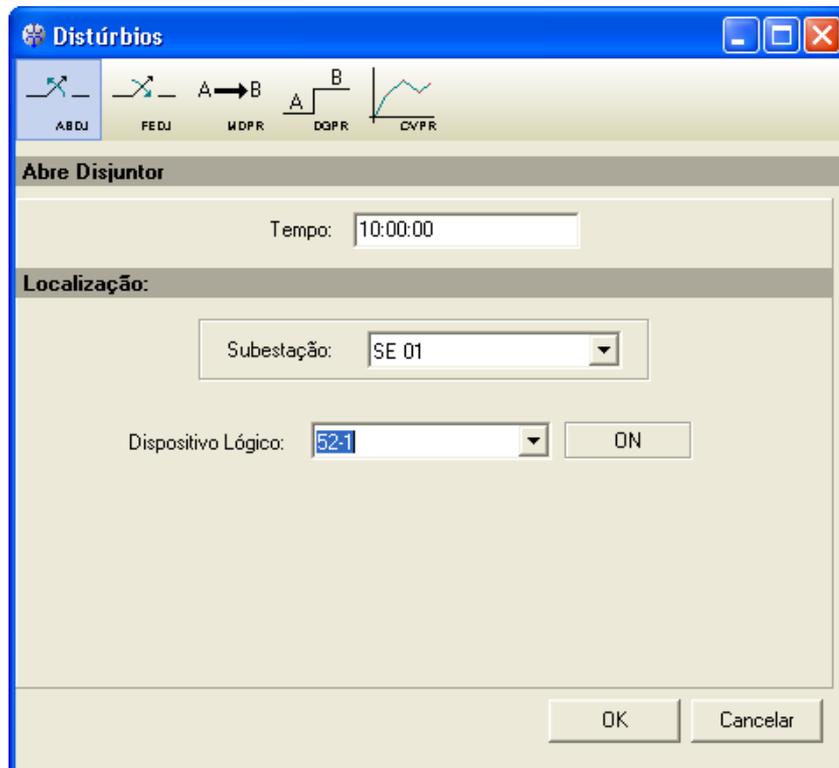


Figura 11: Tela de criação de eventos de abertura de chaves.

Depois de modelado o sistema exemplo, foram realizados testes no servidor OPC implementado. O servidor OPC foi desenvolvido para exteriorizar os valores obtidos pelos medidores alocados no sistema. Como exposto anteriormente, os medidores capturam do simulador, valores de potencia, tensão, posição de TAP de transformadores, estado de chaves e disjuntores. Para testar a interface OPC, foi utilizado um cliente OPC padrão certificado pela OPC Foundation, o Kassl OPC Explorer. Com esse cliente OPC foi aberta uma seção no servidor do PSM e verificada a variação dos valores amostrados a cada segundo e comissionada a escrita nas variáveis. Na figura 12 é apresentada a tela do OPC Explorer responsável por apresentar os valores informados pelo servidor do PSM.

The screenshot shows the dOPC Explorer application window. On the left, a tree view displays the hierarchy: OPC Server > OPC PowerSysLab Server > Server items > Groups > SE01. The main area shows a table of data for the SE01 section. The table has columns for Name, Quality, Time, and Value. The data is as follows:

Name	Quality	Time	Value
dmOPC.SE01.GER.SE01_GER_P	good	09:08:59	-1125
dmOPC.SE01.GER.SE01_GER_Q	good	09:09:59	-123,3403!
dmOPC.SE01.TR1A.SE01_TR1A_2901_SEC	good	09:08:59	1
dmOPC.SE01.TR1A.SE01_TR1A_2901_SEC_CMD	bad	09:00:26	-
dmOPC.SE01.TR1A.SE01_TR1A_5201_DJ	good	05:56:59	1
dmOPC.SE01.TR1A.SE01_TR1A_5201_DJ_CMD	bad	21:00:00	-
dmOPC.SE01.TR1A.SE01_TR1A_2903_SEC	good	05:56:59	1
dmOPC.SE01.TR1A.SE01_TR1A_2903_SEC_CMD	bad	21:00:00	-
dmOPC.SE01.TR1A.SE01_TR1A_2902_SEC	good	05:56:59	0
dmOPC.SE01.TR1A.SE01_TR1A_2902_SEC_CMD	bad	21:00:00	-
dmOPC.SE01.TR1B.SE01_TR1B_2904_SEC	good	05:56:59	1
dmOPC.SE01.TR1B.SE01_TR1B_2904_SEC_CMD	bad	21:00:00	-
dmOPC.SE01.TR1B.SE01_TR1B_5202_DJ	good	05:56:59	1
dmOPC.SE01.TR1B.SE01_TR1B_5202_DJ_CMD	bad	21:00:00	-
dmOPC.SE01.TR1B.SE01_TR1B_2906_SEC	good	05:56:59	1
dmOPC.SE01.TR1B.SE01_TR1B_2906_SEC_CMD	bad	21:00:00	-
dmOPC.SE01.TR1B.SE01_TR1B_2905_SEC	good	05:56:59	0
dmOPC.SE01.TR1B.SE01_TR1B_2905_SEC_CMD	bad	21:00:00	-
dmOPC.SE01.B525P.SE01_B525P_V	good	09:09:59	552,25207
dmOPC.SE01.B525T.SE01_B525T_V	good	05:56:59	0
dmOPC.SE01.SE08.SE01_SE08_2907_SEC	good	05:56:59	1
dmOPC.SE01.SE08.SE01_SE08_2907_SEC_CMD	bad	21:00:00	-
dmOPC.SE01.SE08.SE01_SE08_5203_DJ	good	05:56:59	1
dmOPC.SE01.SE08.SE01_SE08_5203_DJ_CMD	bad	21:00:00	-
dmOPC.SE01.SE08.SE01_SE08_2908_SEC	good	05:56:59	1
dmOPC.SE01.SE08.SE01_SE08_2908_SEC_CMD	bad	21:00:00	-
dmOPC.SE01.SE08.SE01_SE08_2910_SEC	good	05:56:59	0
dmOPC.SE01.SE08.SE01_SE08_2910_SEC_CMD	bad	21:00:00	-
dmOPC.SE01.SE08.SE01_SE08_P	good	09:09:59	1122,5168
dmOPC.SE01.SE08.SE01_SE08_Q	good	09:09:59	703,40447
dmOPC.SE01.TRANS.SE01_TRANS_2911_SEC	good	05:56:59	0
dmOPC.SE01.TRANS.SE01_TRANS_2911_SEC_CMD	bad	21:00:00	-

Figura 12 – Tela do OPC Explorer utilizado para consultar aos valores simulados pelo módulo PSM.

A disposição dos grupos OPC apresentada na figura 12 é um exemplo de como os dados podem ser estruturados na hierarquia criada pelo cliente no servidor OPC. Nesse exemplo, foi criado um grupo para cada subestação modelada no sistema exemplo. Outra possibilidade seria organizar os grupos de itens OPC por nível de tensão, região geográfica ou até mesmo empresas. Essa hierarquização não interfere na hierarquia interna dos dados no servidor OPC, apresentada na seção 4.4, figura 8, pois a hierarquia criada através dos grupos OPC é determinada pelo cliente, podendo coexistir diferentes grupos OPC em um mesmo servidor.

5. Construção do CCM sobre um Software SCADA

Nesse capítulo é apresentado o modelo de CCM (*Control Center Model*) desenvolvido para validar a estrutura de OTS proposta nesse trabalho. Como exposto anteriormente, o módulo CCM será integrado ao PSM via protocolo OPC, onde o CCM será um cliente dos pontos disponibilizados pelo servidor do PSM. Como o propósito desse trabalho é apresentar que o protocolo OPC-DA poderia ser adotado como solução de comunicação de dados em um sistema OTS e que o mesmo possibilitaria que o módulo CCM fosse desenvolvido sobre a mesma plataforma SCADA (*Supervisory Control and Data Acquisition*) utilizada pelo Centro de Controle (COS) a ser replicado, foi criado um exemplo de implementação de CCM sobre uma ferramenta comercial de desenvolvimento de aplicações SCADA, o Elipse E3. Essa ferramenta é desenvolvida pela Elipse Software Ltda. e possibilita que aplicações SCADA sejam desenvolvidas de forma mais eficiente através da utilização de componentes pré-definidos. O Elipse E3 é uma ferramenta genérica de desenvolvimento de softwares de automação, podendo ser programada para atender as necessidades específicas de diferentes processos industriais, porém, sua programação para customizar um aplicativo SCADA de centros de controle de sistemas de energia elétrica é uma tarefa um tanto quanto trabalhosa, devido à complexidade envolvida na operação de tais sistemas.

Como o tema desse trabalho não é a construção de um sistema SCADA e sim uma proposta de integração dos módulos que constituem um sistema OTS, foi decidido que o CCM deveria ser adaptado sobre uma aplicação SCADA existente, pois somente dessa forma seria possível explorar as funcionalidades de um software SCADA real e avaliar a complexidade da integração do módulo CCM sobre uma plataforma existente. Para isto, foi utilizado como plataforma de construção do módulo CCM um programa desenvolvido sobre o Elipse E3 chamado de S4. O S4 foi desenvolvido pela Sul Engenharia e Sistemas Ltda., empresa integradora de sistemas de automação, para realizar funções de supervisão e controle de centros

de operação de sistemas de energia elétrica. Atualmente o S4 encontra-se implantado no centro de controle de três empresas de distribuição de energia elétrica no estado do Rio Grande do Sul.

Nesse capítulo é apresentada uma definição de sistema SCADA, o software Elipse E3, o software S4 e a adaptação dos dados de simulação ao S4.

5.1. Sistemas SCADA

Um sistema SCADA consiste de um conjunto de Unidades Terminais Remotas (UTRs) responsáveis por coletar dados de campo e enviá-los para uma estação mestre através de um sistema de comunicação de dados [48]. A estação mestre possibilita o acesso aos dados via uma IHM (Interface Homem-Máquina) e disponibiliza recursos para realização de operações de supervisão e controle, como por exemplo, um comando de abertura de um disjuntor. A disponibilidade de tais dados permite que a operação do processo seja realizada de forma eficiente, confiável e segura.

Um sistema SCADA pode ser dividido em cinco níveis:

- Dispositivos de controle e instrumentação de campo;
- Conjunto de UTRs;
- Sistema de Comunicação de Dados;
- Estação/Estações Mestre;
- Sistema de processamento, tratamento e armazenamento de dados;
- IHM de supervisão e controle.

A figura 13 apresenta um diagrama esquemático da disposição de cada nível existente em um sistema SCADA.

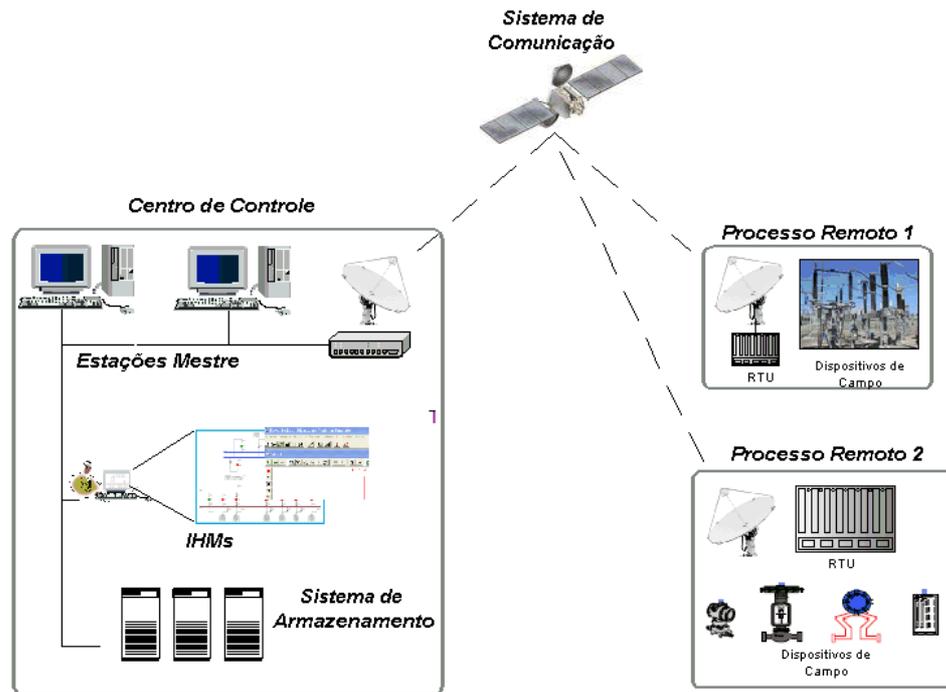


Figura 13: Diagrama esquemático de um sistema SCADA.

As UTRs realizam a interface com os sensores de campo (sensores digitais e analógicos) existentes em cada região de controle remoto. Tais informações são enviadas através de um sistema de comunicação a uma estação mestre. Esse sistema de comunicação pode ser implementado sobre meios físicos de comunicação como par trançado, fibra ótica, rádio, satélite, etc. Protocolos de comunicação específicos são utilizados para realização da transferência de dados de forma eficiente e segura.

A estação mestre reúne os dados provenientes das UTRs e disponibilizam, nas IHMs, as informações necessárias para a supervisão e controle do processo. Em sistemas com presença de muitas UTRs, geralmente são utilizadas mais de uma estação mestre. Também é comum existir redundância dos equipamentos das estações mestre, de forma a aumentar a disponibilidade do sistema.

Também podem fazer parte do sistema SCADA o processamento de alarmes, o registro de eventos e o armazenamento de dados históricos de variáveis digitais e analógicas.

5.2. Elipse E3

O Elipse E3 é um ambiente de desenvolvimento de aplicativos que disponibiliza uma série de recursos que facilitam a implementação de um software SCADA. Dentre as diversas funcionalidades oferecidas pela ferramenta, destacam-se [49]:

- Suporte a diferentes protocolos de comunicação;
- Processamento e visualização de alarmes;
- Conexão a bancos de dados comerciais (Ex.: Oracle[®], Access[®] e MS SQL Server[®]);
- Registro e análise de eventos;
- Armazenamento e análise de histórico de variáveis digitais e analógicas;
- Ambiente de modelagem de bibliotecas para reutilização de código;
- Estrutura Cliente-Servidor;
- Controle de acesso e gerência de contas de usuários;
- Ambiente de desenvolvimento de *scripts* em linguagem Visual Basic;
- Rastreabilidade de código e ações.

Através da utilização das facilidades oferecidas pelo E3, o processo de desenvolvimento da aplicação pode ser realizado por um número menor de pessoas. Além da redução no número de Homens-Hora de um projeto desse porte, a ferramenta também possibilita que pessoas sejam rapidamente treinadas a utilizarem a ferramenta de programação, possibilitando que pessoas menos qualificadas executem tarefas que seriam extremamente complexas sem o auxílio da ferramenta. Isso ocorre principalmente devido ao suporte a diferentes protocolos de comunicação, as ferramentas de conexão e consultas via SQL (*Structured Query Language*) e à arquitetura Cliente-Servidor. Assim, o integrador investe tempo na aplicação final, podendo desenvolver recursos para facilitar e potencializar a interface gráfica do usuário e implementar ferramentas de gerência e manutenção do processo. A seguir é apresentada uma breve descrição dos principais funcionalidades disponibilizadas pelo E3.

5.2.1. Drivers de Comunicação

O E3 permite a comunicação com equipamentos de aquisição de dados (Ex.: CLPs e UTRs), através de drivers de comunicação próprios [49]. Os drivers de comunicação são módulos do E3 disponibilizados por arquivos DLL (*Dynamic Link Library*).

Ao criar um driver de comunicação em uma aplicação, é necessário carregar um arquivo DLL que implemente o protocolo de comunicação específico. Exemplos de protocolos oferecidos pela Elipse são o IEC 60870-5, IEC 61850, DNP3 e ModBus. Uma vez carregada a DLL específica, são oferecidas telas de configuração do driver carregado. Tais telas facilitam a customização do driver, permitindo que parâmetros específicos do protocolo sejam informados, bem como a configuração de parâmetros de endereçamento, time-out, número de tentativas, tempos de varreduras, meio de comunicação (serial, Ethernet), etc.

5.2.2. Alarmes

O E3 dispõe de um conjunto de objetos que permitem ao usuário monitorar alarmes em sua aplicação. Com eles, é possível especificar e gerenciar alarmes e eventos de variáveis de um processo. As condições de alarmes podem ser gerenciadas em diversas fontes. O sistema é composto de um objeto centralizador de alarmes, chamado de servidor de alarmes, de um ou mais objetos de configuração e de um ou mais elementos de visualização de alarmes.

No servidor de alarmes é possível configurar parâmetros que coordenam a gravação, em banco de dados, dos alarmes gerados pela aplicação, bem como consultar informações como número de alarmes ativos, número de alarmes não reconhecidos, etc.

Já no configurador de alarmes é possível configurar cada elemento responsável por monitorar cada ponto de comunicação. Através desse objeto são configurados os tipos de alarmes (digital, analógico, de variação ou de banda morta), seus limites, a área a que pertence o alarme, as mensagens que aparecerão quando o ponto ultrapassar os limites, etc.

O objeto de visualização de alarmes permite definir como as mensagens serão apresentadas ao usuário, cores das mensagens, formas de reconhecimento dos alarmes, filtros, campos de visualização (Data e Hora de entrada em estado alarmado, valor da variável, Nome do ponto de comunicação que gerou o alarme) etc.

5.2.3. Eventos

A ferramenta de eventos trabalha em conjunto com o servidor de alarmes. Todas as variações de estados anormais e normais são gravadas em uma tabela, onde as colunas podem ser customizadas, de forma a oferecer uma história do comportamento do sistema supervisionado. Nessa tabela histórica são gravados, por exemplo, Data/Hora da ocorrência do evento, identificação do evento, o ponto de comunicação que gerou o evento, valor do ponto de

comunicação, Data/Hora do reconhecimento do alarme, usuário que realizou o reconhecimento, etc.

Uma vez que a história dos eventos ocorridos é armazenada em um banco de dados, os mesmos podem ser consultados através de códigos SQL e apresentados em uma tabela através de relatórios customizáveis.

5.2.4. Histórico

O módulo de armazenamento de dados históricos do E3 permite que uma grande quantidade de dados de processo sejam armazenadas de forma rápida, segura e eficiente. Esse módulo parte do princípio de que um dado só deve ser gravado quando houver uma variação significativa (definida pelo usuário) ou quando ultrapassar um intervalo máximo de tempo sem nenhuma gravação (*timeout*). Para isto, é utilizado um algoritmo chamado de BoxCar/BackSlope, que permite uma compactação de até 93% nos dados, com uma recuperação total das informações realmente úteis, dependendo da banda morta definida [49].

Os dados armazenados nas tabelas de registro do histórico podem ser consultados através de consultas SQL e apresentadas em formato tabular ou gráfico. Essas tabelas são armazenadas em banco de dados Oracle ou Microsoft SQL Server, possibilitando que outras aplicações da rede corporativa da empresa acessem os registros gravados pelo SCADA.

5.2.5. Estrutura Cliente Servidor

O E3 implementa o conceito de Domínio, que permite que em um único ambiente, sejam definidas as variáveis do sistema (nomes e endereços de pontos de comunicação), construídas as telas da aplicação, definidos os alarmes e configurados os históricos. Isso permite que toda a configuração da aplicação SCADA seja implementada em um único servidor, possibilitando que uma mesma base de dados seja acessada por diferentes máquinas clientes. Dessa forma, alterações na aplicação SCADA são automaticamente atualizadas nos clientes, pois todos acessam a mesma base de dados. Assim, ao alterar um ponto de comunicação ou um objeto de tela, todos os clientes são atualizados automaticamente.

O controle de acesso a telas e dados se dá através de senhas de usuários e filtros de aplicação. Assim é possível controlar as telas que cada usuário pode acessar, bem como as ações permitidas e os alarmes que podem ser visualizados e reconhecidos.

5.2.6. Biblioteca de Usuário – ElipseX

Os ElipseX são as bibliotecas de objetos do E3 utilizadas para aprimorar a produtividade, manutenção e reaproveitamento de código. Com a utilização dos ElipseX é possível modelar objetos através de propriedades e métodos customizáveis, possibilitando que elementos que necessitam ser instanciados diversas vezes em uma aplicação obedeçam a apenas uma definição. Um exemplo típico de objeto que é instanciado diversas vezes em uma aplicação SCADA de sistemas de energia elétrica é o equipamento disjuntor, onde as propriedades como estado (Ligado/Desligado), comandos, identificação e proteções são comuns a todos os equipamentos desse tipo e podem ser modeladas em apenas uma biblioteca. Assim, ao adicionar uma nova propriedade ou método a esse elemento, todas as instâncias dessa biblioteca específica serão alteradas.

A ferramenta de desenvolvimento de bibliotecas do E3 possibilita a criação de dois tipos de objetos: os XObjects e os XControls.

Os XObjects possibilitam o desenvolvimento de modelos de dados. Tais modelos não apresentam interface gráfica, e são instanciados apenas no servidor. Esses objetos são capazes de concentrar tarefas como tratamento de alarmes, comunicação, armazenamento de dados, etc.

Um exemplo de modelo de dado pode ser um ponto analógico, o qual deve possuir um ponto de comunicação, uma unidade, um tratamento de alarme, o armazenamento em banco de dados, configuração de banda morta, etc.. Outro exemplo de modelo de dado é o próprio disjuntor, que é um equipamento que pode ser modelado como um conjunto de pontos de comando, de pontos de estados, pontos analógicos, alarmes, características elétricas e construtivas, identificação, etc.

Já os XControls possibilitam o desenvolvimento de objetos de telas padronizados. Esses objetos são instanciados nas máquinas clientes e podem acessar dados no servidor através de links a objetos de dados. Dessa forma, a interface gráfica de um elemento como um disjuntor pode ser implementada em uma biblioteca e seu comportamento como: cores para sinalizar o seu estado lógico, alarmes e falha na comunicação são implementados e mantidos em um único local.

Ao modelar os objetos de dados e telas, a produtividade e legibilidade do código aumentam, além de potencializar a aplicação com a extração de diversas informações de forma fácil e eficiente. Além disso, o fato de modelar os elementos que constituem o sistema de automação de forma a replicar seus aspectos físicos e lógicos possibilita que um maior grau de

proximidade entre o ambiente modelado no software e ao mundo real seja obtido, facilitando o desenvolvimento de interfaces mais intuitivas e amigáveis

5.3. Software S4

O S4 é uma aplicação computacional, desenvolvida sobre a plataforma Eclipse E3, que implementa as características necessárias que um software SCADA deve possuir para ser utilizado em um Centro de Operação de Sistema Elétrico (COS). Essa aplicação foi desenvolvida pela Sul Engenharia e Sistemas Ltda. e foi gentilmente cedida para que o estudo objeto desse trabalho fosse elaborado sobre essa aplicação. O S4 consiste de um conjunto de bibliotecas (XControls e XObjects) que facilitam a construção da aplicação SCADA. Além das bibliotecas de dados e telas, o S4 propõe uma forma de organização dos dados internos do E3 e implementa alguns recursos que aprimoram a confiabilidade e a usabilidade do sistema.

Basicamente o S4 é dividido em três camadas, onde o conteúdo de cada camada é apresentado a seguir.

5.3.1. Estruturação dos Drivers de Comunicação

A camada de comunicação é responsável por conter todos os drivers de comunicação da aplicação. Nessa camada são configurados, individualmente, cada driver de comunicação (uma para cada UTR), bem como a definição dos pontos de comunicação que serão tratados pela aplicação. O escalonamento dos pontos, seus significados e unidades de medidas também são definidos nessa camada. A figura 14 apresenta a estruturação de um driver de comunicação responsável por coletar valores de uma determinada UTR existente em uma subestação (Chamada de SE1), através do protocolo IEC 60870-5-101.

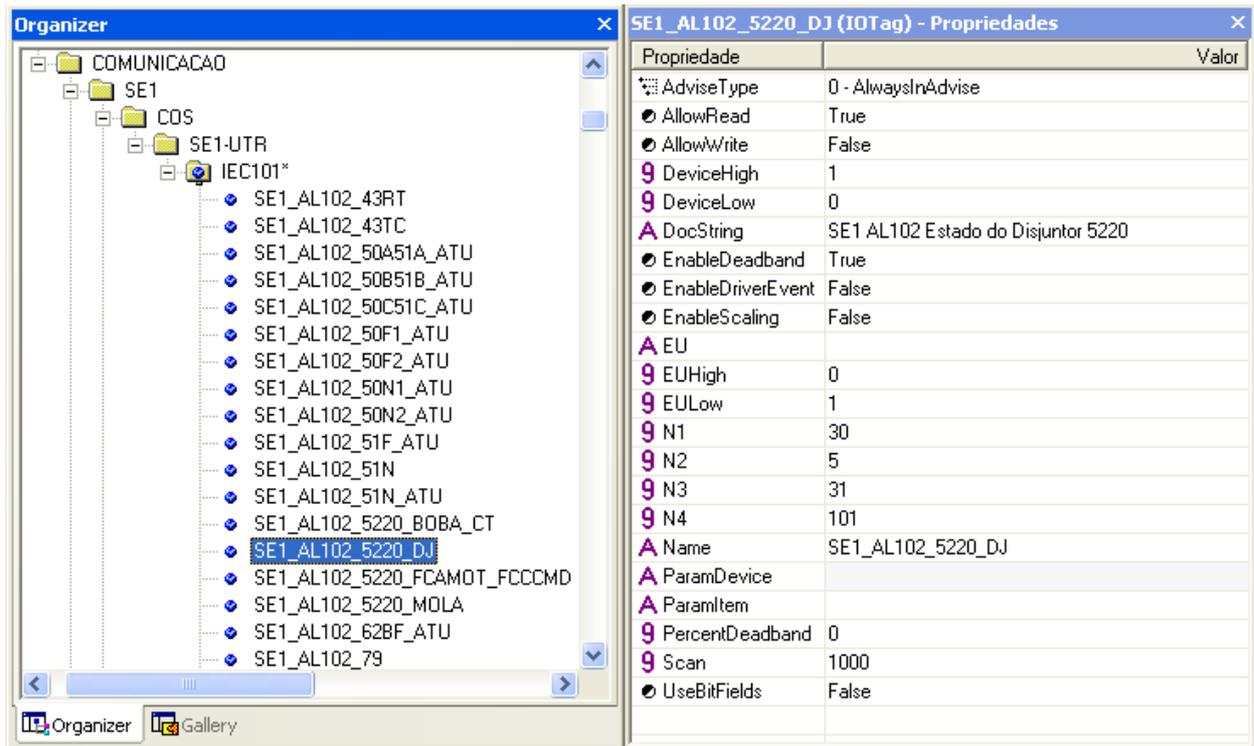


Figura 14: Ambiente de parametrização de um driver de comunicação.

A estruturação do namespace dos pontos de comunicação no S4 utiliza como separador o caracter “_” e os campos são preenchidos por nome da subestação, seguido do nome do módulo e logo após por um complemento que informa se o ponto pertence a um equipamento ou cumpre uma função específica. No exemplo da figura 14, o ponto responsável pela monitoração do estado do disjuntor 5220 do alimentador 102, pertencente à subestação 1, apresenta o nome “SE1_AL102_5220_DJ”. Ao selecionar esse ponto, a janela de propriedades apresenta seus parâmetros específicos como: descrição do ponto (DocString), seus parâmetros de configuração e endereçamento na remota (N1, N2, N3 e N4), a unidade de engenharia (EU) e o escalonamento do ponto para a aplicação (DeviceHigh, DeviceLow, EUHigh, EULow, EnableScaling). Todos os pontos de comunicação são instanciados dentro de um driver de comunicação, no caso da figura 14, representado pela pasta IEC101. Na configuração do driver que são configurados parâmetros específicos do protocolo de comunicação como tempos de varreduras, TimeOuts e endereçamentos como porta e IP.

5.3.2. Estruturação da Camada de Dados

Na camada de dados é introduzido o conceito de XObjects. Nessa camada são modelados os tipos de dados possíveis de serem tratados pela aplicação como pontos analógicos, pontos digitais e pontos de comandos. Nesse nível são configurados limites e mensagens de alarmes, banda morta para armazenamento histórico e relação Valor X Semântica. A figura 15 apresenta a estruturação dos objetos de dados no S4 e as propriedades específicas de um ponto digital.

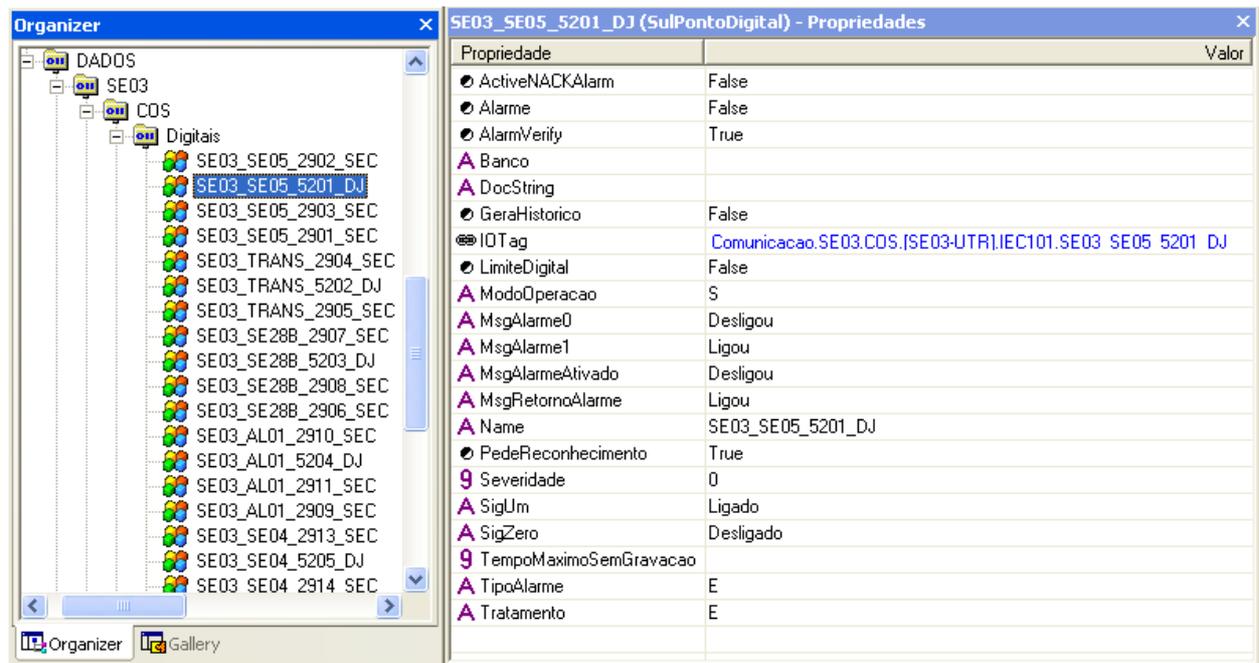


Figura 15: Estruturação dos objetos de dados na aplicação S4.

Na Figura 15 é apresentada a estrutura de pastas onde os dados digitais de uma determinada subestação (SE03) são instanciados. De forma a exemplificar o funcionamento dos XObjects na aplicação, o ponto digital criado para supervisionar o disjuntor 5201, responsável por ligar uma linha de transmissão entre a subestação 3 e a subestação 5, é selecionado. Através da janela de propriedades do objeto é possível visualizar os parâmetros criados para modelar um ponto digital no S4. Propriedades como mensagens de alarme, significado dos valores zero e um do ponto digital e link para o ponto de comunicação.

Nessa camada também são modeladas as estruturas de dados responsáveis por modelar elementos elétricos como transformadores, banco de capacitores, disjuntores, reguladores de tensão, etc.

5.3.3. Telas de Operação do Sistema Elétrico

Por fim, na camada de interface gráfica é introduzido o conceito de XControls. Todos os sinóticos de subestações e representação do sistema elétrico são modelados através da inserção de XControls conectados aos objetos de dados modelados através de XObjects. Dessa forma, evita-se que em tempo de projeto, o programador tenha que criar um grande número de associações à objetos de dados de forma manual. Com os recursos desenvolvidos no S4, o programador tem apenas que inserir um XControl associado a um XObject. A figura 16 apresenta o modelo gráfico de um transformador de três enrolamentos, cujas informações são disponibilizadas por um objeto de dado XObject. Através desse objeto, são exteriorizadas graficamente informações como: características construtivas do equipamento (tensão nominal e esquema de ligação dos enrolamentos), posição do TAP, acesso aos comandos disponíveis no equipamento (alteração da posição do TAP, inclusão de proteções e ventilação forçada), temperatura do óleo, potências ativa e reativa, notas de operação, etc.

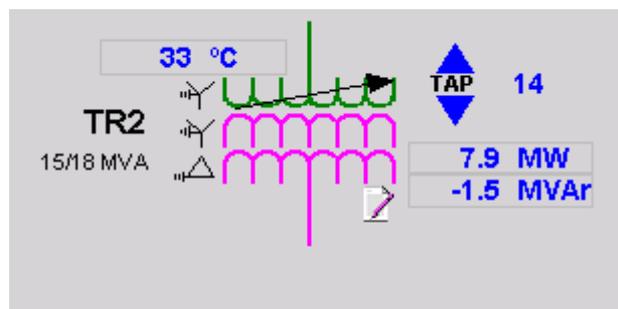


Figura 16: Representação Gráfica de um transformador no S4.

Além dos XControls, nesse nível são também disponibilizados recursos como navegação em dois monitores, navegação pelos elementos do sistema na forma de *TreeView*, telas de gerência de controle de acesso e de configuração de propriedades de dados internos.

5.3.4. Construção do Sistema Exemplo

Para construir o sistema exemplo no CCM, primeiramente foi tomado como base o sistema desenvolvido no módulo PSM, que consiste de um sistema elétrico com 45 barras distribuídas em 28 subestações. Na primeira etapa da construção do CCM no S4, foram

utilizados apenas recursos padrões disponibilizados pelo S4/E3, de forma a construir um sistema SCADA para supervisionar o sistema exemplo de 45 barras. Durante a implementação do exemplo, foi considerado que em cada uma das 28 subestações existiria uma unidade terminal remota com suporte ao protocolo IEC 60870-5-101. Foram implementados os drivers de comunicação, configurados os objetos de dados e customizados os sinóticos de operação, como se fosse supervisionar um sistema real. Como o sistema de 45 barras é um sistema fictício, e obviamente não existem UTRs com suporte a protocolos de comunicação, todos os pontos de comunicação responsáveis por coletar os dados de campo foram substituídos por pontos que geram valores randômicos. A figura 17 apresenta uma tela de visualização do sistema (a), uma tela com o sinótico de uma subestação (b), uma tela com a análise tabular de medidas de um módulo (c) e uma tela de análise histórica de uma variável analógica (d).

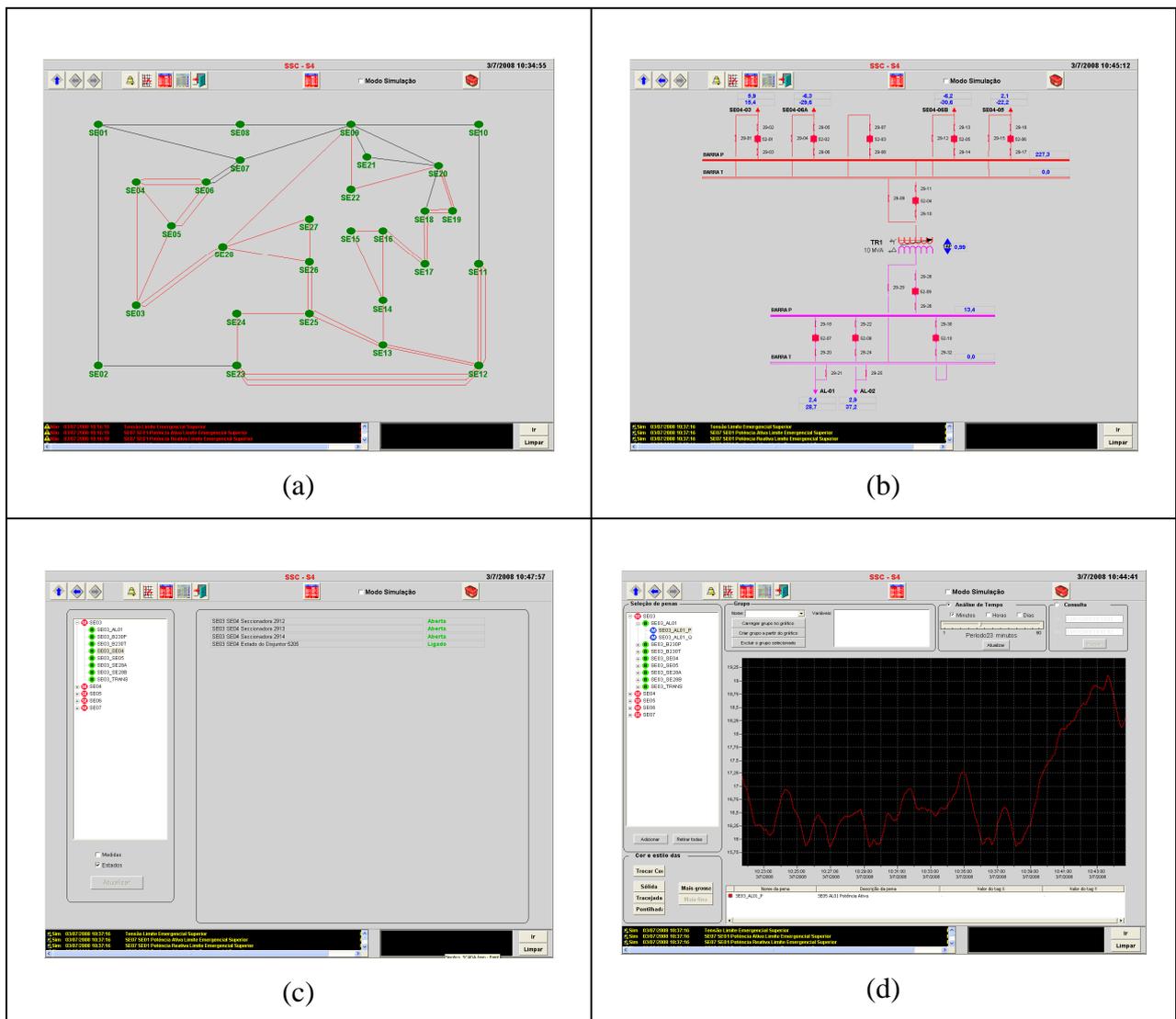


Figura 17: Implementação do sistema exemplo de 45 barras sobre o S4.

A intenção de se construir um sistema SCADA de forma mais próxima possível à encontrada nos centros de operação foi realizada para avaliar exatamente as dificuldades que seriam encontradas para adaptá-la de forma a cumprir as tarefas do módulo CCM.

5.4. Adaptação do Cliente OPC ao módulo SCADA/CCM

Depois de construído o sistema exemplo sobre o E3 / S4, foram analisadas formas de se integrar os dados provenientes do PSM à estrutura existente. Imaginando que a implantação da estrutura proposta fosse realmente implementada em um centro de controle, a adaptação do módulo de simulação não poderia interferir nas lógicas e estruturas de dados utilizadas para controlar o sistema elétrico. Uma vez que o centro de controle de uma empresa que trabalha com energia elétrica é o principal agente para garantir a segurança operativa do sistema [9], é inconcebível que um módulo de simulação de treinamento interferira na confiabilidade de tal sistema. Dessa forma, a adaptação dos dados provenientes do PSM no sistema SCADA foi realizada de forma incremental, ou seja, sem alterar a estrutura existente, apenas adicionando novas estruturas de dados capazes de interagir com o módulo PSM.

A primeira etapa desse processo foi a construção de um módulo cliente OPC no sistema SCADA. Como o S4 divide claramente a camada de comunicação de dados, foi possível adaptar o cliente OPC exatamente como se fosse uma nova remota do sistema, criando um cliente OPC para cada subestação. Os pontos de comunicação em cada cliente OPC foram criados automaticamente com o auxílio das ferramentas de OPC Browser disponibilizadas pela tecnologia OPC-DA. Como a estrutura hierárquica de dados no servidor OPC do módulo PSM subdivide as subestações em nós diferentes (figura 8), a importação dos dados para cada subestação foi ágil e segura, garantindo que pontos de diferentes subestações não fossem misturados. A figura 18 apresenta o driver OPC configurado para coletar os dados de simulação referentes a subestação 1.

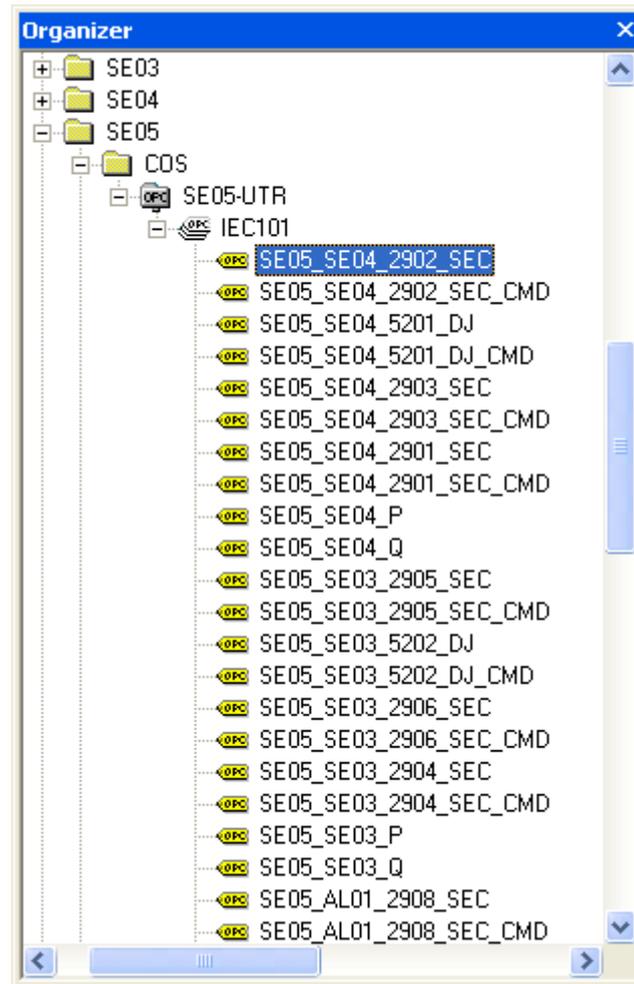


Figura 18: Pontos OPC criados em um cliente dentro do sistema SCADA.

Para manter a estruturação das pastas de comunicação criadas no S4, foi criado um grupo OPC contendo o mesmo nome do protocolo utilizado pela remota responsável por coletar os pontos na subestação real. Com isso, a referência dos objetos de dados aos pontos de simulação segue a mesma estrutura adotada para o sistema real. Conforme será apresentado a seguir.

Com a estrutura de comunicação semelhante à utilizada para controlar um sistema real, a etapa seguinte foi adaptar a estrutura dos pontos OPC ao projeto criado para o sistema exemplo. Para isto, bastou adicionar os drivers a uma pasta chamada de “Comunicação_Sim” e incluí-la ao projeto de comunicação anteriormente criado.

O passo seguinte foi em adaptar os objetos de dados implementados através das bibliotecas de XObjects do S4. Para isso, foi realizada uma cópia de toda a pasta de dados do projeto original, a qual armazena todas as instâncias de XObjects, renomeando de “Dados”, para

“Dados_Sim”. Os primeiros objetos de dados adaptados foram os que modelam os pontos digitais. Estas bibliotecas padronizam a semântica de valores, o armazenamento histórico, o estado anormal e as mensagens de alarmes. A adaptação dos objetos de dados para receberem as informações de simulação foi realizada através do redirecionamento dos ponteiros dos pontos de comunicação (tags de comunicação), substituindo o link anterior (que apontava para um driver de comunicação responsável por coletar os valores de uma UTR) para o link com o driver cliente OPC (driver responsável por coletar os valores disponibilizados pelo PSM). Essa substituição foi automatizada através da utilização de ferramentas de edição do tipo (Localizar/Substituir), pois como a comunicação OPC foi estruturada de forma idêntica a comunicação utilizada para comunicar com as UTRs, a única diferença entre os ponteiros estava no fato de que as UTRs foram instanciadas em uma pasta chamada de “Comunicacao” e os pontos OPC em uma pasta chamada de “Comunicacao_Sim”. As figuras 19 e 20 apresentam, respectivamente, as propriedades de um objeto de dado digital que tem como fonte de informação um ponto de comunicação proveniente de uma UTR e um objeto de dado digital que tem como fonte de informação um ponto simulado pelo PSM.

Propriedade	Valor
<input checked="" type="checkbox"/> ActiveNACKAlarm	False
<input checked="" type="checkbox"/> Alarme	False
<input checked="" type="checkbox"/> AlarmVerify	True
<input type="checkbox"/> Banco	
<input type="checkbox"/> DocString	
<input checked="" type="checkbox"/> GeraHistorico	False
<input checked="" type="checkbox"/> IOTag	Comunicacao.SE03.COS.[SE03-UTR].IEC101.SE03_AL01_5204_DJ
<input checked="" type="checkbox"/> LimiteDigital	False
<input type="checkbox"/> ModoOperacao	S
<input type="checkbox"/> MsgAlarme0	Desligou
<input type="checkbox"/> MsgAlarme1	Ligou
<input type="checkbox"/> MsgAlarmeAtivado	Desligou
<input type="checkbox"/> MsgRetornoAlarme	Ligou
<input type="checkbox"/> Name	SE03_AL01_5204_DJ
<input checked="" type="checkbox"/> Pedereconhecimento	True
<input type="checkbox"/> Severidade	0
<input type="checkbox"/> SigUm	Ligado
<input type="checkbox"/> SigZero	Desligado
<input type="checkbox"/> TempoMaximoSemGravacao	
<input type="checkbox"/> TipoAlarme	E
<input type="checkbox"/> Tratamento	E

Figura 19: Propriedades de um objeto de dado digital apontando para um tag de comunicação proveniente de uma UTR.

SE03_AL01_5204_DJ (SulPontoDigital) - Propriedades	
Propriedade	Valor
<input checked="" type="checkbox"/> ActiveNACKAlarm	False
<input checked="" type="checkbox"/> Alarme	False
<input checked="" type="checkbox"/> AlarmVerify	True
<input type="checkbox"/> Banco	
<input type="checkbox"/> DocString	
<input checked="" type="checkbox"/> GeraHistorico	False
<input checked="" type="checkbox"/> IOTag	Comunicacao Sim.SE03.CDS.[SE03-UTR].IEC101.SE03_AL01_5204_DJ
<input checked="" type="checkbox"/> LimiteDigital	False
<input type="checkbox"/> ModoOperacao	S
<input type="checkbox"/> MsgAlarme0	Desligou
<input type="checkbox"/> MsgAlarme1	Ligou
<input type="checkbox"/> MsgAlarmeAtivado	Desligou
<input type="checkbox"/> MsgRetornoAlarme	Ligou
<input type="checkbox"/> Name	SE03_AL01_5204_DJ
<input checked="" type="checkbox"/> PedirReconhecimento	True
<input type="checkbox"/> Severidade	0
<input type="checkbox"/> SigUm	Ligado
<input type="checkbox"/> SigZero	Desligado
<input type="checkbox"/> TempoMaximoSemGravacao	
<input type="checkbox"/> TipoAlarme	E
<input type="checkbox"/> Tratamento	E

Figura 20: Propriedades de um objeto de dado digital apontando para um tag de comunicação proveniente do módulo PSM.

O mesmo procedimento de adaptação foi realizado para os objetos de dados analógicos e de comandos. Para adaptar os objetos de dados responsáveis pela modelagem de equipamentos e módulos, foi necessário apenas modificar as cláusulas informadas à ferramenta de edição de “Localizar /Substituir”. Como os modelos de objetos de equipamentos e módulos não utilizam como fonte de informação pontos de comunicação, e sim outros objetos de dados, bastou substituir nos ponteiros de fonte de dados dos objetos a palavra “Dados.” pela e palavra “Dados_Sim.”. As figuras 21 e 22 apresentam respectivamente, as propriedades de um objeto de dado responsável por modelar um alimentador contendo informações reais, e um objeto de dado modelado contendo informações provenientes do sistema de simulação.

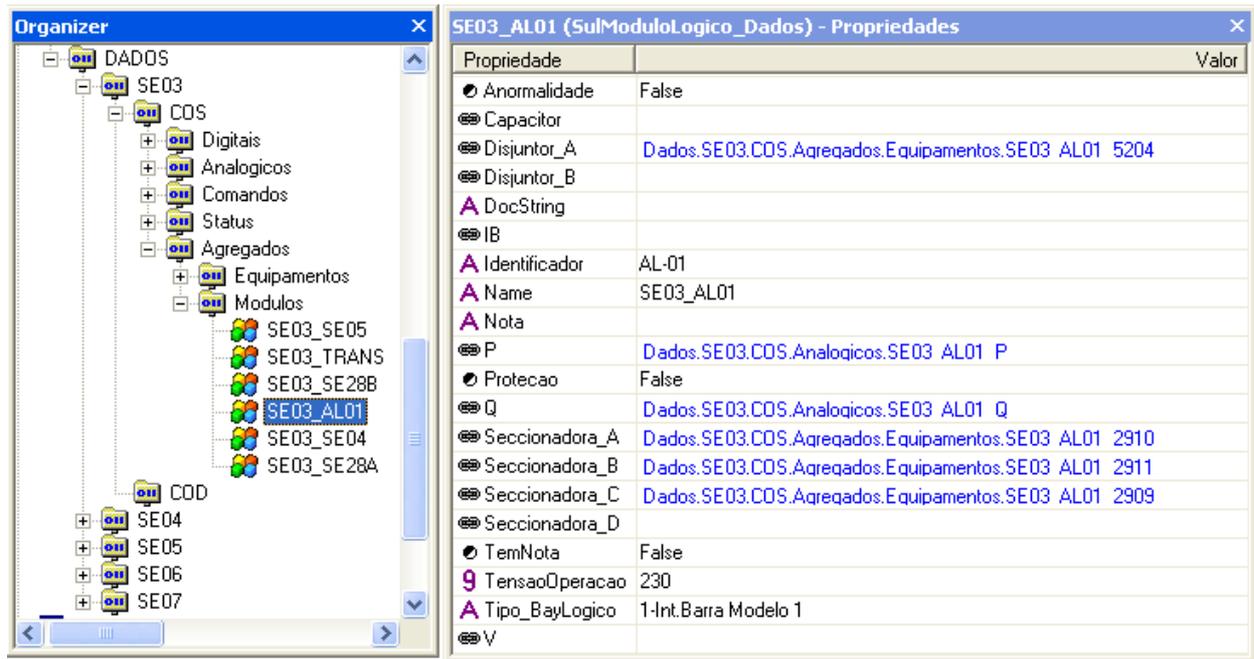


Figura 21: Propriedades de um objeto de dados responsável por modelar um módulo de alimentador contendo informações reais.

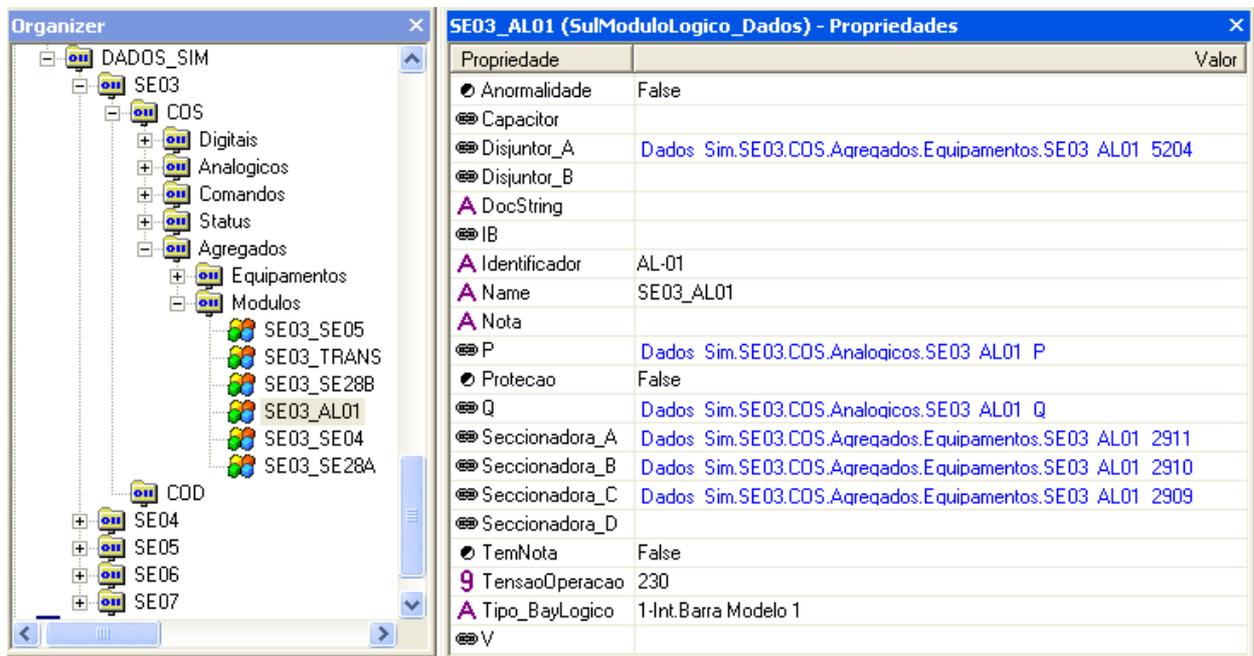


Figura 22: Propriedades de um objeto de dados responsável por modelar um módulo de alimentador contendo informações simuladas.

Com os objetos de dados estruturados e prontos para serem utilizados pela interface gráfica, foram copiadas todas as telas implementadas para operar o sistema exemplo. Todas as telas copiadas foram renomeadas de forma a possuir o sufixo ”_Sim”. Novamente a ferramenta de edição “Localizar/Substituir” foi utilizada para alterar os ponteiros dos objetos de tela, fazendo com que os mesmos apontassem para objetos instanciados na pasta de dados “Dados_Sim”. As cores de fundo das telas de simulação foram alteradas de forma a exteriorizar a informação de que a mesma está apresentando valores através do módulo CCM e não do sistema SCADA real. As figuras 23 e 24 apresentam, respectivamente, o diagrama de operação responsável pelo controle e supervisão de uma subestação monitorada por uma UTR e o diagrama de operação do módulo CCM.

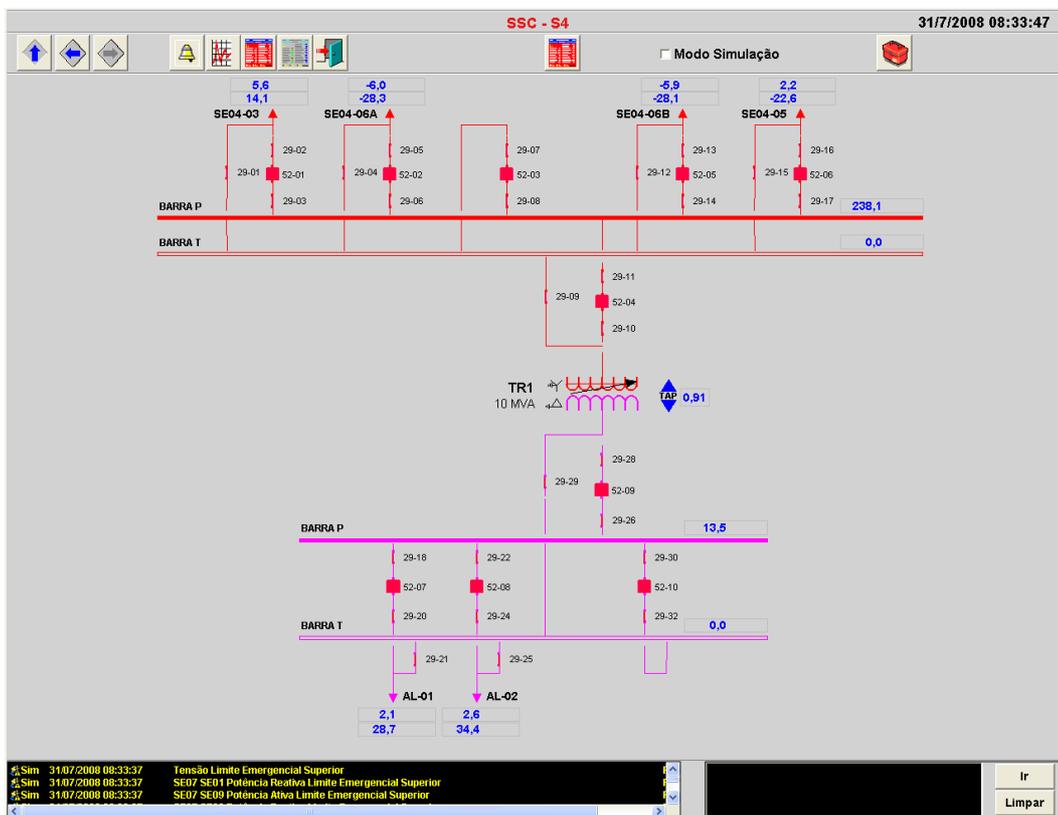


Figura 23: Diagrama de operação de uma subestação monitorada por uma UTR.

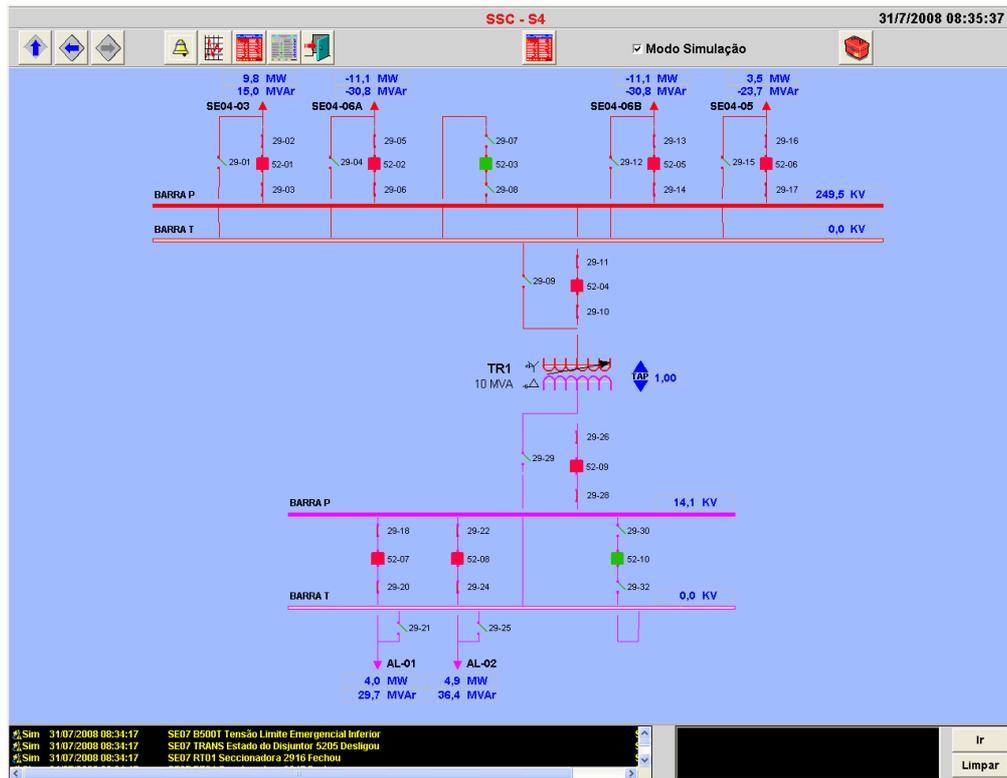


Figura 24: Diagrama de operação de uma subestação simulada pelo módulo PSM e acessada via CCM.

Todas as implementações realizadas sobre a plataforma SCADA não acarretaram em alterações dos dados internos da aplicação de supervisão e controle do sistema real. Isso é importante para evitar que o módulo CCM comprometa ou altere o correto funcionamento do SCADA que já se encontra em operação no centro de controle. Porém apenas uma mudança teve que ser implantada na navegação entre telas, pois agora a interface SCADA permite comutar das telas de operação para as telas de simulação. Essa alteração foi implementada na barra de navegação e permite que a qualquer momento o operador altere o modo de operação.

6. Integração dos Módulos ES, PSM, SCADA e CCM

Esse capítulo apresenta as diferentes arquiteturas que podem ser implementadas pelo OTS proposto nesse trabalho, a estrutura de rede de computadores utilizada para avaliar o funcionamento do OTS, a configuração necessária de firewall e componentes OPC/DCOM em cada computador.

6.1. Possíveis Arquiteturas para Distribuição dos Processos em Rede

Devido à grande versatilidade oferecida pelo protocolo OPC e as características Cliente/Servidor do Elipse E3, um grande número de possibilidades de estrutura de computadores em rede podem ser adotadas para a construção do OTS. São apresentadas três diferentes formas de se integrar os módulos PSM, ES, CCM e SCADA.

6.1.1. Stand-Alone

Na configuração Stand-Alone, todos os módulos são instalados em um mesmo computador, não sendo necessária a configuração da comunicação em rede. As duas aplicações (Elipse E3/S4 e o PSL-SDK) são instaladas em uma mesma máquina, possibilitando que pequenos sistemas sejam simulados e/ou supervisionados. A vantagem desse tipo de estrutura está na facilidade de configuração e manutenção, sendo uma boa alternativa para os casos de auto-treinamento. As desvantagens desse tipo de configuração estão relacionadas às limitações de processamento, da pouca similaridade com o centro de controle e do número de usuários

conectados ao sistema. A figura 25 apresenta a configuração da plataforma configurada para trabalhar na forma Stand-Alone.

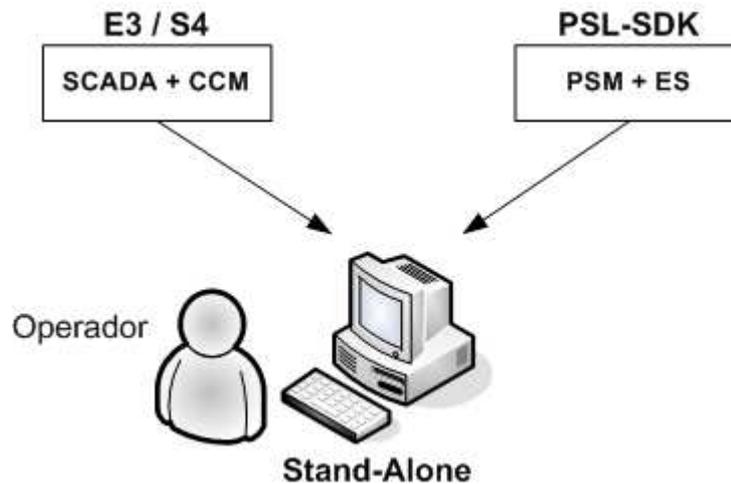


Figura 25: Plataforma configurada para trabalhar no modo Stand-Alone.

6.1.2. Distribuído em dois Servidores e Diversos Clientes

Nessa arquitetura, os módulos PSM e ES, desenvolvidos sobre o PSL-SDK, são instalados em um servidor específico, responsável apenas pela simulação e disponibilização dos valores via OPC Server. Os módulos CCM e SCADA, desenvolvidos no Elipse E3/S4, são instalados em um servidor responsável por coletar os valores de UTRs, estabelecer comunicação com o PSM e disponibilizar acesso aos diferentes clientes que acessam as telas de operação. Dessa forma é possível compartilhar entre os operadores, tanto a operação do sistema real quanto o ambiente de simulação e treinamento. Outra vantagem está na divisão dos processamentos de simulação e monitoração/controle, aumentando a capacidade global de processamento do sistema. A desvantagem está no maior investimento devido à infra-estrutura de rede e de computadores envolvida. A figura 26 apresenta a configuração da plataforma distribuída em dois servidores e possibilitando a conexão de diversos operadores.

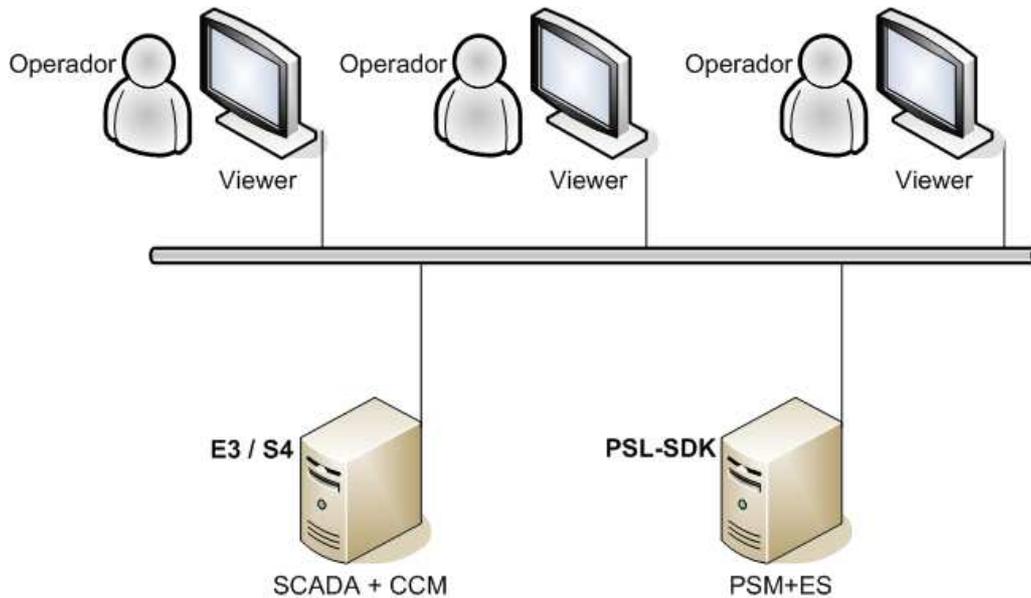


Figura 26: Plataforma configurada para trabalhar distribuída em dois servidores.

6.1.3. Distribuído em Diversos Servidores e Diversos Clientes

O Eclipse E3 disponibiliza um recurso chamado de Domínio Distribuído, que permite que as tarefas executadas pelo E3 Server sejam distribuídas em mais de uma máquina. Com a utilização desse recurso, as tarefas de SCADA e CCM podem ser desacopladas em servidores diferentes, fazendo com que o sistema de supervisão e controle não tenha seu desempenho afetado pela inserção dos dados de simulação na plataforma S4. A desvantagem desse tipo de estrutura, quando comparada à anterior, está relacionada à complexidade da implementação e manutenção da plataforma, bem como no custo de aquisição da infra-estrutura. A figura 27 apresenta a configuração da plataforma distribuída em três servidores e possibilitando a conexão de diversos operadores.

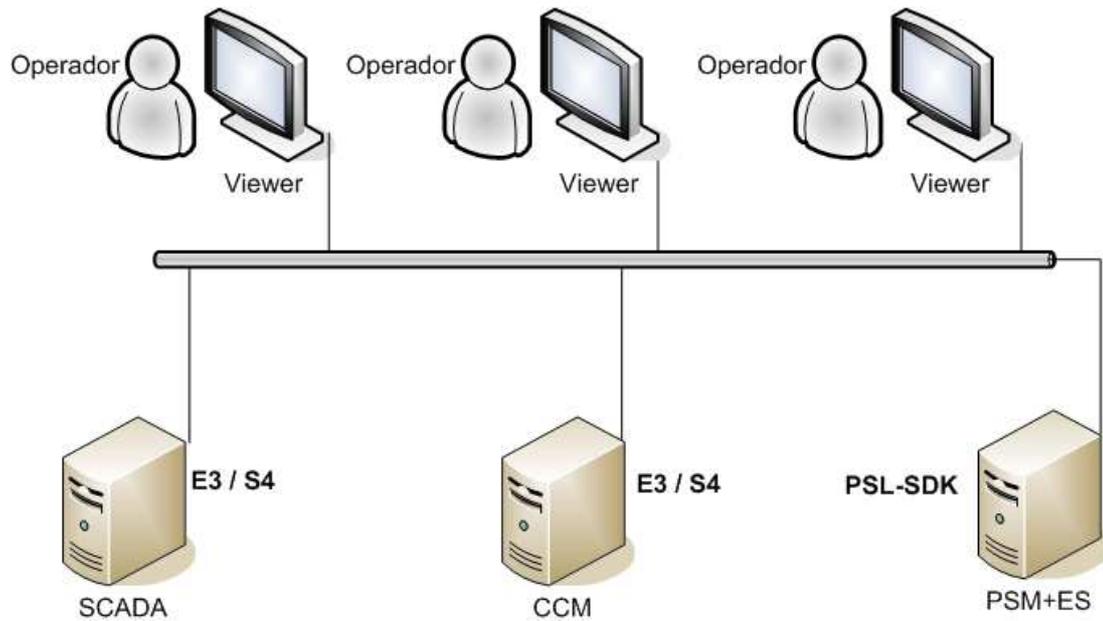


Figura 27: Plataforma configurada para trabalhar distribuída em três servidores.

6.2. Estrutura de Rede Utilizada para Testes e Avaliação

Para testar a aplicação desenvolvida e avaliar o desempenho e as possibilidades de treinamento disponibilizadas pela plataforma proposta, foi utilizada uma configuração semelhante à apresentada no item 6.2. Para isto, foram utilizados dois servidores: um servidor para o módulo CCM/SCADA, instalando-se o Eclipse E3, as bibliotecas S4 e o sistema exemplo desenvolvido; e outro servidor para a plataforma PSL-SDK responsável pelos módulos PSM e ES. Como terminais de operação foram utilizados dois computadores para executar a aplicação Eclipse E3 Viewer. Todos os computadores foram conectados a uma rede local Ethernet 100 Mb. Em todas as máquinas foi instalado o sistema operacional Microsoft Windows XP Service Pack 2.

6.3. Configuração do Microsoft Windows XP Service Pack 2 para Liberação do OPC/DCOM

O Service Pack 2 do Windows XP disponibiliza significativos avanços desenvolvidos para proteger os diferentes meios passíveis de sofrerem ataque de usuários não autorizados à dados de segurança do sistema operacional. Nessa atualização foi disponibilizado um software firewall integrado ao sistema operacional e alterados os comportamentos dos objetos COM e DCOM [50]. Essas atualizações fazem com que algumas medidas devam ser tomadas para que a comunicação OPC em rede funcione corretamente. A seguir são apresentadas as configurações de firewall e liberação de serviços de componentes necessárias para que a comunicação OPC entre os servidores CCM e PSM seja estabelecida.

6.3.1. Configuração do Windows Firewall

Existem duas formas de evitar que o software Windows Firewall não interfira no funcionamento da comunicação OPC entre os servidores. A primeira é desabilitar o firewall, o que pode acarretar em um comprometimento da segurança do sistema no caso de uma rede não confiável. A segunda forma é fazer com que o Firewall permita exceções. Tais exceções devem ser configuradas na guia “Exceptions”. Basicamente, deve-se informar as portas TCP e UDP de número 135, estas utilizadas pelo DCOM, além da porta TCP de número 6515, responsável pela conexão dos terminais de operação ao servidor E3, conforme apresentado na figura 28. Para habilitação do OPC, devem ser informados os programas que utilizam tal comunicação, no caso, o Elipse E3 em um servidor e o PSL-SDK no outro. Feito isto, a aplicação está pronta para funcionar com o Firewall habilitado.

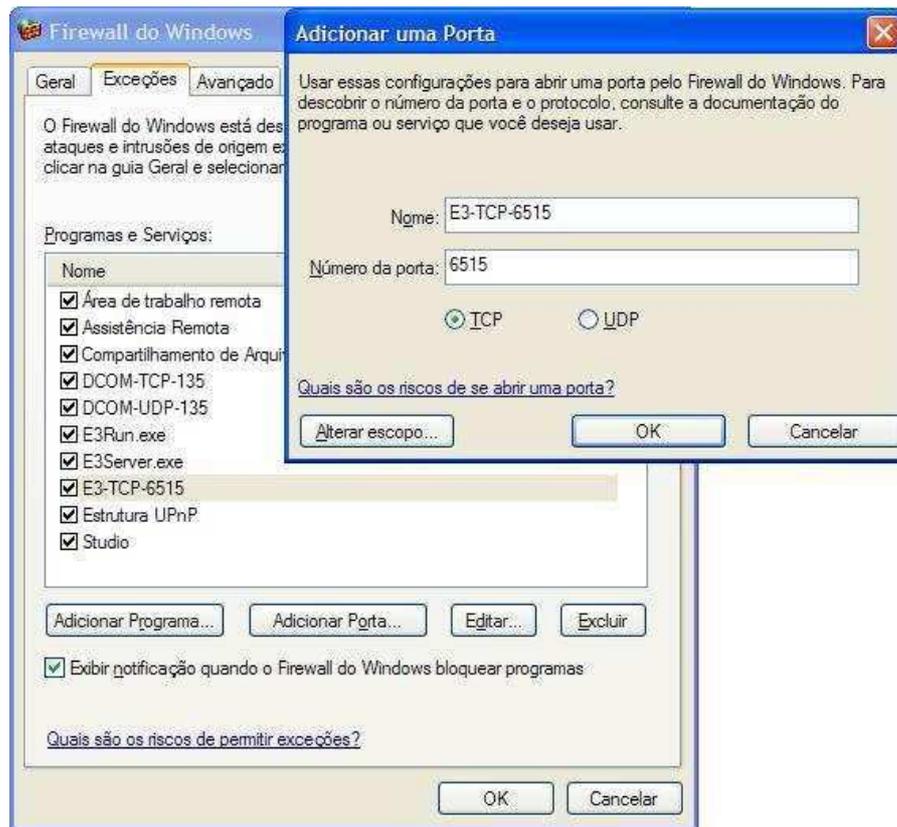


Figura 28: Inclusão de Exceções no Windows Firewall.

6.3.2. Configuração do DCOM

As modificações no DCOM do Service Pack 2 foram feitas para contornar falhas de segurança evidenciadas com a proliferação de ataques de *worms* e vírus baseados neste protocolo de comunicação, sem prejudicar aplicações que necessitam efetivamente do DCOM. Um exemplo da evolução do DCOM no lançamento do Service Pack 2 é que as versões mais antigas do aplicativo servidor COM restringiam que um aplicativo fosse executado apenas localmente sem ser exposto na rede por meio do DCOM. Quando os usuários têm acesso a um aplicativo servidor COM, eles têm acesso tanto ao uso local como remoto [51]. Essas características foram restringidas no Service Pack2 e para que o sistema OTS seja executado em rede, algumas liberações devem ser realizadas nos *Serviços de Componentes* do sistema operacional. Através do serviço de componentes devem ser configuradas as permissões de acesso, inicialização e ativação remota de objetos COM pelos usuários habilitados. Também via *Serviço de Componentes* devem ser habilitados os serviços DCOM das aplicações que podem ser acessadas,

inicializadas e ativadas remotamente. A figura 29 apresenta a tela de serviços de componentes utilizada para configurar as habilitações do serviço DCOM da aplicação E3Server.

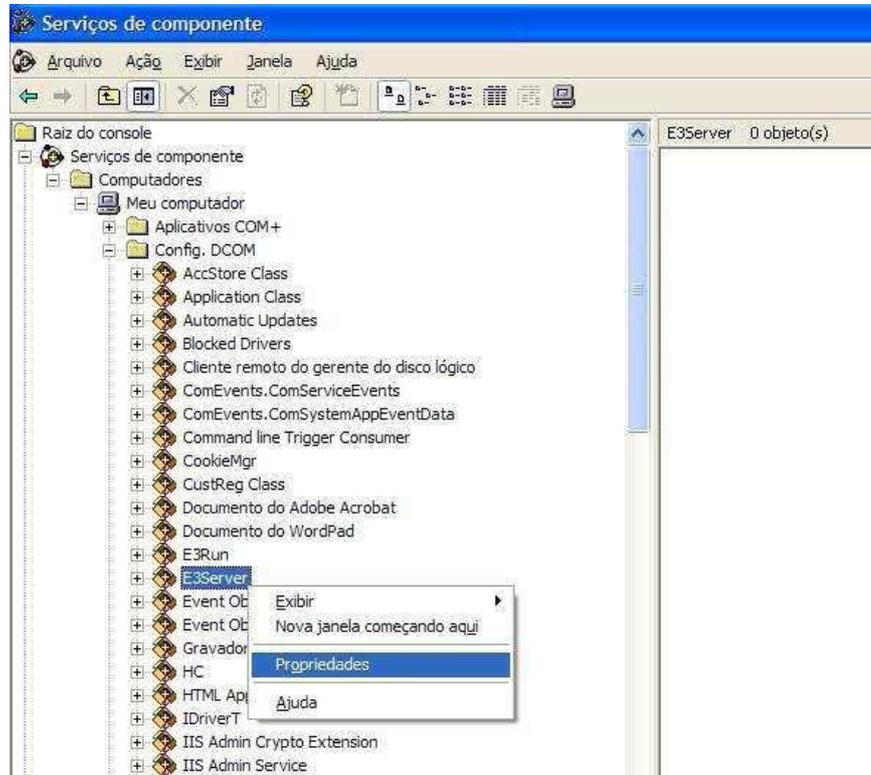


Figura 29: Tela de configuração de serviços DCOM.

6.4. Operação Integrada dos Módulos

Após habilitados os computadores que fazem parte do sistema OTS e configurado o cliente OPC do Elipse E3 para acessar as informações do módulo PSM via rede, o sistema SCADA+CCM pode ser inicializado. Ao inicializá-lo, a aplicação PSM é ativada remotamente. A partir desse momento é disponibilizado acesso para os terminais de operação, que podem operar tanto o sistema SCADA quanto o PSM via CCM. No caso da necessidade de configuração de um cenário específico no módulo PSM, o mesmo pode ser configurado via ES em tempo de execução pelo instrutor ou pelo próprio operador. Isso pode ser realizado a qualquer momento sem que os módulos SCADA e CCM sejam afetados.

7. Avaliação e Interpretação dos Resultados

Nesse capítulo serão apresentadas as diferentes formas de utilização da plataforma desenvolvida e como estas podem ser utilizadas como ferramenta de suporte ao aprimoramento técnico dos operadores sob treinamento. Primeiramente serão apresentadas as características do sistema configurado na forma stand-alone, seguindo das características do sistema quando configurado para suportar funções de treinamento supervisionado e por fim as facilidades oferecidas pelo modo de suporte à simulação em tempo real.

7.1. Modo Stand-Alone

No modo stand-alone, como exposto em 6.1.1, todas as aplicações são instaladas na mesma máquina. Através da utilização do módulo ES, o usuário pode carregar um cenário previamente criado para estudo, criar um cenário totalmente novo ou reconstruir um cenário passado a partir de um banco de dados que contém o histórico de variáveis do sistema elétrico como injeção de potências, tensões, estados de dispositivos lógicos, posição de tap de equipamentos de controle de tensão, etc. Com o cenário definido, o mesmo é carregado no módulo PSM, podendo assim inicializar a simulação que executa o algoritmo de fluxo de potências de forma cíclica. Com o ambiente de simulação configurado, o usuário pode abrir o módulo CCM.

Através do módulo CCM o operador pode se familiarizar com a navegação no software SCADA, acessar valores digitais e analógicos, visualizar os registros de valores históricos, tratamento de alarmes, filtragem de mensagens de anormalidade e ferramentas de análise de eventos. Outra funcionalidade a ser explorada são as características de simulação, que possibilitam que o operador execute ações básicas de comando como abrir e fechar dispositivos lógicos (chaves e disjuntores), alterar posição de tap de transformadores, analisar influencia de

elementos reativos como reatores e capacitores e verificar o comportamento temporal do sistema elétrico quando submetido a diferentes curvas de carregamento.

Procedimentos operativos adotados pela empresa para realizar tarefas como transferência de módulos de carga, transferência de linhas de transmissão e isolamento de dispositivos para manutenção podem ser exercitados livremente pelo operador. Reconstituição de cenários passados, onde diversos eventos foram registrados, podem ser analisados detalhadamente, possibilitando uma maior compreensão do comportamento elétrico da empresa e avaliando as ações tomadas e as possíveis alternativas para solucionar o problema ocorrido.

7.2. Treinamento Supervisionado

No modo de treinamento supervisionado existe a presença de um instrutor, o qual é responsável por criar cenários operativos, elaborar scripts de eventos, avaliar ações individuais de cada operador e verificar a capacidade da equipe em resolver situações complexas onde a segurança operativa do sistema depende de um trabalho de equipe eficiente. O papel de instrutor pode ser elaborado por uma equipe de pessoas experientes e com grande conhecimento sobre as características do sistema elétrico e dos procedimentos operativos adotados pela empresa.

Esse modo de treinamento permite que uma equipe de operadores seja treinada em conjunto, fazendo com que não apenas a capacitação técnica da equipe seja exercitada, mas também suas habilidades de comunicação, pró-atividade, trabalho em equipe, liderança, comprometimento com o grupo e tomada de decisões sob pressão psicológica. Como existe a presença de instrutores experientes conduzindo o treinamento, um grande número de situações vivenciadas no passado podem ser replicadas em um curto período de tempo e de forma segura, facilitando a transferência de conhecimento e agilizando o processo de aprimoramento do quadro de funcionários da empresa.

Para avaliar o desempenho do trabalho em equipe do grupo de operadores o instrutor pode submeter cenários que demandem que tarefas distintas sejam executadas de forma simultânea e sincronizadas, em diferentes regiões de operação. Assim, é possível avaliar como cada operador resolve o problema em sua área de operação de forma sincronizada com o trabalho de outros operadores responsáveis por regiões adjacentes. Também pode ser avaliada a transferência de regiões de operação entre operadores, pois na ocorrência de condições climáticas adversas em determinadas regiões pode fazer com que uma região tenha que ser

operada por um número maior de operadores, fazendo necessária a migração de operadores de regiões mais calmas para regiões mais conturbadas.

Após cada tarefa proposta, o instrutor pode analisar uma lista de eventos contendo o histórico das ações do treinamento, possibilitando a identificação de ações incorretas. Com base no relatório o instrutor pode sugerir outras formas de resolução do problema do cenário proposto. O mesmo cenário pode ser submetido diversas vezes, possibilitando ao operador o privilégio de errar e corrigir sua ação, o que no sistema real é inadmissível.

7.3. Simulação de Ações em Tempo Real

Outra forma de treinamento possibilitada pela estrutura proposta nesse trabalho é a possibilidade de utilizar a plataforma de simulação para executar um cenário capturado diretamente do sistema SCADA. Dessa forma, as últimas medidas coletadas pelo sistema de supervisão e controle (também chamado de *snapshot*) alimentam o construtor de cenário, disponibilizando no ambiente de simulação o estado atual do sistema elétrico. Assim, caso o operador queira simular um determinado procedimento antes de realmente executá-lo na rede real, basta que o operador troque de modo de operação SCADA para modo de operação Simulação. No momento em que o operador troca de modo de operação, a cor de fundo da tela de operação é alterada e todas as variáveis supervisionadas têm como fonte de dados o servidor OPC do módulo PSM e não mais as diversas remotas do sistema SCADA.

Com a disponibilização desse módulo no centro de controle, o operador do sistema pode verificar se a manobra a ser executada violará algum limite operacional. Com isso, evita-se que proteções como de sobre-corrente atuem durante um procedimento de transferência de carga, ou até mesmo se as tensões nas barras do sistema ficarem dentro dos patamares especificados e que nenhum dispositivo terá suas capacidades nominais violadas.

8. Conclusão

Durante o desenvolvimento deste trabalho, foi possível avaliar alguns fatores que motivaram a pesquisa no assunto. O objetivo principal do trabalho era avaliar os benefícios e deficiências de um sistema OTS onde o módulo CCM pudesse ser implementado através do mesmo sistema SCADA utilizado pelo centro de controle. Para realizar tal tarefa decidiu-se utilizar o protocolo padrão de comunicação entre processos chamado de OPC-DA. A escolha do OPC-DA se mostrou eficiente em resolver os problemas básicos levantados no início da pesquisa como:

- Um grande número de sistemas SCADA comerciais possuem suporte ao protocolo;
- Fácil implementação;
- Protocolo mundialmente aceito pelo mercado de automação;
- Protocolo com capacidade de ser integrado a uma rede de computadores baseada em TCP-IP;
- Permitir que seja implementada uma estrutura Cliente-Servidor, onde uma mesma base de dados seja acessada por um número indeterminado de clientes;
- Desempenho na exteriorização dos dados, não sendo o gargalo do processo de simulação/exteriorização das medidas fornecidas pelo módulo OTS.

Realmente o protocolo OPC-DA atendeu todos os requisitos especificados durante a etapa de planejamento da arquitetura e permitiu construir uma interface de comunicação no módulo PSM que possibilita que qualquer cliente OPC-DA acesse os resultados e controle a simulação em andamento, possibilitando que um grande número de aplicações SCADA, independente do fabricante, acesse sua base de dados e aproveite seus recursos internos para construir o módulo CCM. Porém, é importante ressaltar que foram verificados alguns pontos de deficiência que devem ser considerados antes de adotá-lo como solução final para resolver o problema de comunicação entre os módulos PSM e CCM:

- Dificuldades na parametrização de permissão de usuários dos componentes DCOM quando o sistema é utilizado em rede.
- Configuração das exceções nas regras dos Firewalls ativos na rede de interligação PSM-CCM;
- Podem surgir problemas de registro do Windows, uma vez que esse é essencial para o correto funcionamento dos componentes COM/DCOM e consecutivamente do OPC;
- Apesar de existirem componentes que adaptem o COM/DCOM em outros sistemas operacionais, a utilização do OPC está muito ligada ao sistema operacional Windows.

Mesmo diante das deficiências expostas acima, o OPC-DA se mostrou uma boa solução de comunicação entre os módulos PSM e CCM.

Com relação às características gerais do sistema OTS construído, concluiu-se que o maior ganho em se implementar a solução proposta está no fato do operador poder utilizar a mesma interface do sistema SCADA da empresa para ser treinado utilizando os recursos de OTS. Essas características fazem com que a solução proposta não somente permita o treinamento do operador com relação ao comportamento físico do sistema elétrico, mas também os recursos disponíveis no sistema SCADA, tais como: alarmes, comandos, lista de eventos, etc. Outra possibilidade disponibilizada pela estrutura proposta seria a característica da mesma poder ser integrada como um módulo de simulação em tempo de operação, possibilitando que o operador simule um determinado procedimento operativo antes de executá-lo.

Durante o processo de implementação do sistema exemplo, verificou-se que a maior dificuldade envolvida na configuração do sistema OTS está em garantir a integridade entre o modelo do sistema elétrico implementado no módulo PSM e os diagramas de operação existentes no sistema SCADA. Essa dificuldade é resultante, principalmente, pela falta de uma base cadastral padronizada e única, onde ambas as aplicações, PSM e SCADA, pudessem acessar a mesma fonte de informação. Acredita-se que a utilização da norma IEC 61970 pode contribuir para facilitar o processo de implantação e garantir uma maior confiabilidade para o sistema OTS.

Com a implementação do OTS finalizada, ficou evidente as potencialidades de treinamento oferecidas pela ferramenta e o quanto a implantação de tal sistema pode colaborar no aprimoramento e capacitação técnica tanto da equipe de operadores como da equipe de planejamento de procedimentos operativos. Além disso, o processo de treinamento com o auxílio

do OTS pode proporcionar um ambiente de troca de experiências e debates sobre os procedimentos adotados pela empresa e sobre o comportamento físico do sistema em que trabalham diariamente. Como resultado direto de tal capacitação, pode-se afirmar que operadores devidamente treinados diminuem o risco de operações indevidas e se sentem mais preparados para agir em situações de recomposição do sistema elétrico, reduzindo assim a frequência e o tempo das interrupções de fornecimento de energia elétrica. Com isso, a empresa melhora seus índices de qualidade de fornecimento e a sociedade pode contar com um mais serviço eficiente de fornecimento de energia elétrica.

8.1. Perspectivas Futuras

O trabalho apresentado descreve uma proposta de desenvolvimento de um sistema OTS aberto, capaz de ser acoplado a diferentes CCMs devido à utilização do protocolo padrão OPC-DA. Durante a elaboração do trabalho foram verificadas algumas oportunidades de desenvolvimento que fariam com que a estrutura proposta apresentasse maior confiabilidade, um maior número de funcionalidades e um menor esforço de implantação.

A principal dificuldade constatada foi que o modelo elétrico da rede a ser simulada pelo PSM e os diagramas unifilares de operação do sistema SCADA não utilizam uma mesma base de dados. Assim, os unifilares de operação do sistema SCADA podem representar um sistema diferente do simulado pelo módulo PSM, fazendo com que os resultados da simulação sejam avaliados de forma equivocada. Outra consequência indesejável seria a manutenção do sistema, pois se uma nova subestação for inserida, tanto o SCADA quanto o PSM terão que ser atualizados individualmente. Para solucionar esse problema, acredita-se que o estudo e a aplicação do conjunto de normas IEC 61970 [52], [53] e [54] ajudem a incrementar o sistema proposto nesse trabalho.

O conjunto de normas IEC 61970 foi criado para definir uma API (*Application Program Interface*) de forma a padronizar a integração de informações entre aplicações que constituem um EMS (*Energy Management System*). Esse padrão foi baseado no documento CC-API (EPRI Control Center API) desenvolvido pelo EPRI (*Electric Power Research Institute*) durante o projeto de pesquisa RP-3654-1. Segundo [52], os principais objetivos do EPRI CC-API foram:

- Reduzir o custo e o tempo necessário para adicionar novas aplicações a um EMS;

- Proteger o investimento realizado nas aplicações e sistemas existentes que formam um EMS.

A tarefa do projeto CC-API foi determinar os requisitos para possibilitar que um sistema EMS seja composto por diferentes aplicações, desenvolvidas por diferentes fornecedores, ou até mesmo, possibilitar a integração do sistema EMS a outros sistemas como o de gerenciamento da distribuição ou da geração de energia elétrica. As padronizações dessas APIs permitem que diferentes aplicações e sistemas acessem uma mesma base pública única e troquem informações entre si independentemente de como essas informações são representadas internamente. Basicamente, acredita-se que a utilização das normas IEC 61970 – 301 [52], IEC 61970 – 501 [53] e IEC 61970 – 453 [54] ajudariam significativamente na resolução dos problemas encontrados e na consolidação do ambiente proposto como parte integrante de um EMS. Essas normas tratam respectivamente da modelagem das informações, como estas informações poderiam trafegar entre aplicações e sobre como as informações gráficas poderiam ser compartilhadas entre diferentes aplicações que constituem um EMS.

Outra oportunidade verificada durante a elaboração do trabalho está relacionada ao número de seções de simulação suportadas pelo módulo PSM. Atualmente o módulo executa apenas uma seção de simulação e essa característica pode resultar em restrições de utilização, principalmente no modo de simulação de tempo real, onde mais de um operador pode requisitar diferentes simulações ao mesmo tempo.

Também foi verificado que a inserção de informações como lógicas de intertravamentos e parametrização dos dispositivos de proteção podem incrementar a potencialidade da ferramenta durante o processo de treinamento. Acredita-se que tais informações poderão ser integradas automaticamente ao sistema de cadastro através de arquivos XML (eXtensible Markup Language) que poderão ser importados de subestações automatizadas obedecendo a norma IEC 61850.

Referências Bibliográficas

- [1] www.aneel.gov.br
- [2] www.ons.org.br
- [3] www.ccee.org.br
- [4] J.G. Waigth, K. Nodehi, A. Bose, E. Dobrowolski, R. Wasley, “Transportable Simulator Trains Control Center Operators”. IEEE Computer Applications in Power, Vol. 5, No.2, pp 23-28. April 1992.
- [5] S. Lee, “Instructor Guidelines for Use of an Operator Training Simulator”, Technical Review, EPRI, October 2001.
- [6] E. Handschin and Petroianu, “Energy Management Systems”. Springer-Verlang. 1991.
- [7] S. Gissinger, P. Chaumès, J. P. Antoine, A. Bihain, M. Stubbe, “Advanced Dispatcher Training Simulator”, IEEE Computer Applications in Power, Vol 13, No. 2, pp. 25-30, April 2000.
- [8] J. J. R. de Oliveira, L. C. Lima, C. A. da Silva Neto, L. A. C. Pereira, R. B. Sollero, C. R. Leite, O. F. Ricciari Junior, H. A. R. Volskis “Sistema para Treinamento e Certificação de Operadores no Ambiente SAGE”. V Simpase - Simpósio de Automação de Sistemas Elétricos. Recife, Pernambuco. 2003.

- [9] F. Maghsoodlou, R. Masiello, T. Ray, "Energy Management System", IEEE Power & Energy Magazine, Vol.3, pp. 49-57 September/October 2004

- [10] H.-J. Appelpath, A. Lucks, T. Luhmann, T. Schmedes, M. Uslar, L. Winkels "Interaction of EMS related systems by using the CIM standard". Available online from Langdale Consultants at <http://www.langdale.com.au/styler/xpetal/>

- [11] W. Xingping, Z. Yang, W. Xiwei "A New Generation EMS". Proceedings of the International Conference on Power System Technology - PowerCon 2002, IEEE Publishing, 2002.

- [12] T. Dy-Liacco, "Modern Control Centers and Computer Networking", IEEE Computer Applications in Power, Vol 7, N0 4, pp. 17-22. October 1994.

- [13] S. Lutterodt, Y. Logeay, R. Knoepfel, R Skiold, "Improving Human Performance in the Control Center". CIGRE WG 39.03; ELECTRA No. 174, pp. 90-105. October 1997.

- [14] P.Sigari, M. Rafian-Naini, K.J. Sluis, "Portable and Affordable Operator Training Simulators", IEEE Computer Applications in Power, Vol. 06, No. 3, pp. 39-44. July 1993.

- [15] S. Almeida, N. Machado, and R. Pestana, "Voltage Collapse: Real Time and Preventive Analysis in the Portuguese Transmission System". IEEE Lausanne PowerTech 2007, pp.1817-1822. July 2007.

- [16] G. Zhang, A. Bose, "Scenario Building for Operator Training Simulators Using a Transient Stability Program", IEEE Transactions on Power Systems, Vol. 4, No. 4, pp. 1542-1549. October 1989.

- [17] T. Kobayashi, D. Moridera, S. Fukui, K. Komai, "Verification of an Advanced Power System Restoration Support System Using an Operator Training Simulator". IEEE Transactions on Power Systems, Vol. 9, No. 2, pp. 707-713. May 1994.
- [18] M. M. Eissa, Z. Chaczko, E. Alrahali, "Interactive Protection System for Operator Training Simulator Using New Wavelet Techniques". The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007).
- [19] K. I. Geisler, S. A. Neumann, K. L. Costin, P. K. Bower "Model and Product Based Integrated Systems for Utility Operations". IEEE: Computer Applications in Power, Vol. 5, No. 3, pp 15-20. July 1992.
- [20] G. P. Azevedo, B. Feijó, M. Costa, "Control Centers Involve with Agent Technology". IEEE Computer Application in Power, Vol. 13, No. 3, pp. 48-53. July 2000.
- [21] R. Talbot, P. Robison, "Integration Enhances Control Room Operations". IEEE Computer Application in Power, Vol. 6, No.1, pp. 10-15. January 1993.
- [22] R. Podmore, "Common Information Model - a Developer's Perspective". Proceedings of the 32nd Hawaii International Conference on System Sciences, IEEE Publishing, 1999.
- [23] D. Shi, Y. Lee, X. Duan, Q.H. Wu "Power System Data Warehouses". IEEE Computer Application in Power, Vol. 14, No. 3, pp 49-45. July 2001.
- [24] D. Becker, H. Falk, J. Gillerman, S. Mauser,, R.Podmor, L. Schneberger, "Standards-Based Approach Integrates Utility Applications", IEEE Computer Applications in Power, Vol.13, pp.13-20, Oct. 2000
- [25] www.opcfoundation.org
- [26] <http://www.opcfoundation.org/Products/ProductVendors.aspx>

- [27] OPC Task Force , OLE for Process Control – OPC, www.opcfoundation.org
- [28] Frank Iwanitz, Jürgen Lange “OLE for Process Control – Fundamentals, Implementation, and Application”. Hüthing Verlag Heidelberg, 2001
- [29] Don Box, “Essential COM”. Addison-Wesley, 1998
- [30] “OLE for Process Control (OPC) for new Industrial Automation Systems”, Y. Shimanuki, Proceedings of 1999 IEEE Conference on Systems, Man, and Cybernetics, Vol. 6, pp.1048-1050. 1999.
- [31] A. Chisholm, “DCOM, OPC and Performance Issues”. OPC Foudation – www.opcfoundation.org, White paper, 1998.
- [32] P. Bhatt, R. P. Baldevia, “Integrate IEDs With OPC Technology”. Power Systems Conference: Advanced Metering, Protection, Control, Communication, and Distributed Resources, pp. 391-407. March 2006.
- [33] V. V. Tan, D-S. Yoo, and M-J. Yi, “Modern Distributed Data Acquisition and Control Systems Based on OPC Techniques”. Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07), 2007.
- [34] X. Hong, W. JianHua, Z. ShiQuan, “The Distributed Data Integration and Performance Evaluation in Power Automation System”. Proceedings of PDCAT 2003 - The Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies, pp. 468-471. August 2003.
- [35] Tuan Dang, “Integration of Power Plant Information System with Business Information System in the Open Eletricity Market: Challenges and Solutions”, IEEE 2007

- [36] T. Berry, B. Shephard, “Information Exchange Standards for Power System Monitoring and Control”. Power System Management Control, pp. 139-145. April 2002.
- [37] R. A. Santos, J. E. Normey-Rico, A. M. Gómez, L. F. A. Arconada, C. de Prada Moraga “OPC based distributed real time simulation of complex continuous processes”. Simulation Modelling Practice and Theory, Vol.13, No.7, pp. 525 - 549. October 2005.
- [38] M. Johnstone, D. Creighton, S. Nahavandi “Enabling Industrial Scale Simulation / Emulation Models”. Winter Simulation Conference, 2007, pp. 1028-1034. December 2007.
- [39] R. F Chu, E. J. Dobrowolski, E. J. Barr, J. McGeehan, D. Scheurer, and K. Nodehi, “Restoration Simulator Prepares Operators for Major Blackouts”. IEEE Computer Applications in Power, Vol. 4, No. 4, pp. 46-51. October 1991.
- [40] “OPC DA 2.00 Specification”, OPC Foundation, www.opcfoundation.org.
- [41] “OPC AE 1.10 Specification”, OPC Foundation, www.opcfoundation.org.
- [42] “OPC HDA 1.20 Specification”, OPC Foundation, www.opcfoundation.org.
- [43] “OPC XMLDA 1.01 Specification”, OPC Foundation, www.opcfoundation.org.
- [44] “XML Specification”, W3C, <http://www.w3.org/XML>.
- [45] Rogério Amorim de Faria, “XML Treinamento avançado em XML”. Editora Digerati Books, 2005
- [46] “Agencia Nacional de Energia Elétrica – ANEEL” - Resolução Normativa N° 158, De 23 De Maio De 2005, Pg 2, Art. 2°, Modulo 3

- [47] Alessandro Manzoni, “Desenvolvimento de um Sistema Computacional Orientado a Objetos para Sistemas Elétricos de Potência: Aplicação a Simulação Rápida e Análise da Estabilidade de Tensão”. Tese de Doutorado, Programa de Pós-Graduação em Engenharia Elétrica - COPPE/UFRJ. Março de 2005.
- [48] Gordon Clarke and Deon Reynders, “Modern SCADA Protocols”. Newnes-Elsevier, 2004.
- [49] “Manual do Usuário Elipse E3”, Elipse Software Ltda., disponível em www.elipse.com.br, Porto Alegre – RS, 01/12/2005, Pg93.
- [50] “Using OPC via DCOM with Microsoft Windows XP Service Pack 2”, version 1.10, OPC Foundation, Karl-Heinz Deiretsbacher, Jim Luth. OPC, www.opcfoundation.org.
- [51] “Configurando o Windows XP SP2, Windows 2003 Server SP1 e R2 para aplicações Elipse”. Diogo Augusto Hiller, www.elipse.com.br, 24/10/2006.
- [52] IEC 61970, “Energy management system application program interface (EMS-API) - Part 301: Common Information Model (CIM) base”, International Standard IEC, 2005.
- [53] “Energy management system application program interface (EMS-API) - Part 501: Common Information Model Resource Description Framework (CIM RDF) schema”, International Standard IEC, 2006.
- [54] “Energy management system application program interface (EMS-API) - Part 453: CIM based graphics exchange”, International Standard IEC, 2008.