

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - UTFPR
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO
ESPECIALIZAÇÃO EM ENGENHARIA DE SOFTWARE

MARCELO PECENIN

**MODELO DE ARQUITETURA BASE PARA IMPLEMENTAÇÃO DE
APLICAÇÕES BASEADAS NA IEC-61850**

MONOGRAFIA DE ESPECIALIZAÇÃO

MEDIANEIRA

2011

MARCELO PECENIN

**MODELO DE ARQUITETURA BASE PARA IMPLEMENTAÇÃO DE
APLICAÇÕES BASEADAS NA IEC-61850**

Monografia apresentada como requisito parcial
à obtenção do título de Especialista na Pós
Graduação em Engenharia de Software, da
Universidade Tecnológica Federal do Paraná -
UTFPR - Câmpus Medianeira.

Orientador: Prof. Me. Alan Gavioli

MEDIANEIRA

2011

TERMO DE APROVAÇÃO

Modelo de Arquitetura Base para Implementação de Aplicações Baseadas na IEC-61850

por

Marcelo Pecenin

Esta monografia foi apresentada às 10:20 h do dia 10 de dezembro de 2011 como requisito parcial para a obtenção do título de Especialista no curso de Especialização em Engenharia de Software, da Universidade Tecnológica Federal do Paraná, Câmpus Medianeira. Os acadêmicos foram argüidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Me. Alan Gavioli
UTFPR - Câmpus Medianeira
(orientador)

Prof. Me. Claudio Bazzi
UTFPR - Câmpus Medianeira

Prof. Me. Fernando Schutz
UTFPR - Câmpus Medianeira

Às memórias de Teresinha Berta Pecenin, que
sempre incentivou minha educação formal.

AGRADECIMENTOS

A Jesus Cristo, por ter me dado saúde e força.

Ao professor, mestre e orientador Alan Gavioli, pelo apoio e orientação na elaboração deste trabalho.

À minha família, pela confiança e apoio durante toda esta jornada.

Aos professores do curso de Especialização em Engenharia de Software, por todos os ensinamentos passados.

A Itaipu Binacional pelo apoio financeiro.

A todos amigos e colegas que de alguma maneira contribuíram para o desenvolvimento deste trabalho.

“Os tolos e os fanáticos estão sempre
seguros de si, mas os sábios são
cheios de dúvidas.”

RESUMO

PECENIN, Marcelo. Modelo de Arquitetura Base para Implementação de Aplicações Baseadas na IEC-61850. 2011. 62 f. Monografia (Especialização em Engenharia de Software). Universidade Tecnológica Federal do Paraná, Medianeira, 2011.

Os sistemas de automação de subestação têm como principais objetivos a proteção, controle e supervisão dos sistemas de distribuição de energia elétrica. Esses sistemas estão se tornando cada vez maiores, mais dinâmicos e distribuídos, entretanto, a diversidade de soluções baseadas em *software* e protocolos proprietários de cada fabricante, muitas vezes incompatíveis entre si, apresentam dificuldades para a expansão e evolução desses sistemas. A padronização tornou-se então uma palavra chave para o desenvolvimento da conectividade e interoperabilidade entre sistemas e equipamentos de automação de subestação. Nesse contexto, organizações de padronização como o EPRI e a IEC elaboraram, após uma década de trabalho, a norma IEC-61850, que define um conjunto de requisitos técnicos necessários para permitir que sistemas e equipamentos de diferentes fabricantes e gerações tecnológicas possam se comunicar. A norma aplica modelos de dados orientados a objetos para descrever propriedades e funcionalidades para serem implementadas e controladas, definindo também um conjunto de serviços para interação entre dispositivos, porém, os modelos são definidos de maneira abstrata e não focam em detalhes de implementação inerentes ao processo de desenvolvimento. Este trabalho apresenta uma proposta de estruturação e modelagem da arquitetura base para desenvolvimento de sistemas de automação de subestação compatíveis com a norma IEC-61850, com foco em detalhes e aspectos não especificados na normativa. A modelagem proposta foi elaborada utilizando um conjunto de diagramas da linguagem UML. Os diagramas elaborados permitiram identificar e analisar aspectos que não foram detalhados na norma, promovendo discussões sobre a viabilidade de implementação e possíveis aprimoramentos futuros da modelagem proposta.

Palavras-chave: Sistemas de Automação de Subestação. Modelagem UML. IEC-61850. IED.

ABSTRACT

PECENIN, Marcelo. Base Model Architecture for Implementation of Applications Based on IEC-61850. 2011. 62 f. Monografia (Especialização em Engenharia de Software). Universidade Tecnológica Federal do Paraná, Medianeira, 2011.

The substation automation systems have as main objectives the protection, control and supervision of the distribution systems of electric power. These systems are becoming larger, more dynamic and distributed, however, the diversity of solutions based on proprietary software and protocols of each manufacturer, often incompatible with each other, present difficulties for the expansion and evolution of these systems. The standardization then became a keyword for the development of connectivity and interoperability between systems and substation automation equipment. In this context, standards organizations such as IEC and EPRI developed, after a decade of work, the IEC-61850, which defines a set of technical requirements to allow systems and equipment from different manufacturers and technology generations communicate with each other. The standard applies object-oriented data models to describe properties and functionalities to be implemented and controlled, also defining a set of services for interaction between devices, however, the models are defined in an abstract way and not focus on implementation details related to the development process. This monograph presents a proposal for structuring and modeling of the base architecture for the development of substation automation systems compliant with IEC-61850, with a focus on details and aspects not specified in the standard. The proposed model was developed using a set of UML diagrams. The elaborated diagrams allowed to identify and analyze issues that were not detailed in the standard, promoting discussions on the feasibility of implementation and possible future enhancements of the proposed model.

Keywords: Substation Automation Systems. UML Modeling. IEC-61850. IED.

LISTA DE FIGURAS

1	Conversão Analógica/Digital	18
2	Esquema de Comunicação entre Nós Lógicos	27
3	Barramentos de Comunicação e Níveis Hierárquicos	28
4	Estrutura Lógica de Comunicação	30
5	Representação UML da Linguagem SCL	30
6	Estrutura de Arquivos da Linguagem SCL	31
7	Representação Gráfica do Modelo de Dados de um IED	32
8	Estrutura de Comunicação ACSI	34
9	Protocolos e Perfis de Comunicação	36
10	Estrutura de Comunicação em uma Subestação Elétrica	39
11	Modelo Conceitual da Norma IEC-61850	40
12	Estrutura Lógica de Componentes de um IED	41
13	Estrutura de Pacotes para Código Fonte e Componentes	42
14	Dependências dos Processos e <i>Threads</i> de um IED	44
15	Dependências dos Processos e <i>Threads</i> para Disparo de Eventos	46
16	Diagrama de Classe para Criar uma Nova Associação Orientada à Conexão	47
17	Diagrama de Sequência para Criar uma Nova Associação Orientada à Conexão	48
18	Diagrama de Classe para Executar um Serviço da Interface ACSI	49
19	Diagrama de Sequência para Executar um Serviço da Interface ACSI	50
20	Diagrama de Classe para Disparar e Registrar um Evento GOOSE	52
21	Diagrama de Sequência para Disparar e Registrar um Evento GOOSE	53
22	Diagrama de Classe para Processar um Evento Enviando uma Mensagem GOOSE	54

23	Diagrama de Sequência para Processar um Evento Enviando uma Mensagem GOOSE	55
24	Diagrama de Classe para Receber e Processar uma Mensagem GOOSE	56
25	Diagrama de Sequência para Receber e Processar uma Mensagem GOOSE . .	57

LISTA DE TABELAS

1	Série de Documentos que definem a Norma IEC-61850	26
2	Finalidade dos Arquivos da Linguagem SCL	31
3	Grupos de Nós Lógicos	33
4	Tipos de Mensagens e Classes de Desempenho	35

LISTA DE SIGLAS

ACSI	<i>Abstract Communication Service Interface</i>
ADC	<i>Analog-to-Digital Converter</i>
CID	<i>Configured IED Description</i>
DAC	<i>Digital-to-Analog Converter</i>
DSP	<i>Digital Signal Processing</i>
EPRI	<i>Electric Power Reserch Insitute</i>
F	<i>Function</i>
FAT	<i>Factory Acceptance Test</i>
GOOSE	<i>Generic Object Oriented Substation Event</i>
GSSE	<i>Generic Substation Status Event</i>
ICD	<i>IED Capability Description</i>
IEC	<i>International Electrotechnical Commission</i>
IED	<i>Intelligent Electronic Device</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IHM	<i>Interface Homem Máquina</i>
ISO	<i>International Organization for Standardization</i>
LC	<i>Logical Connection</i>
LN	<i>Logical Node</i>
MMS	<i>Manufacturing Message Specification</i>
OSI	<i>Open Systems Interconnection</i>
PC	<i>Physical Connection</i>
PD	<i>Physical Device</i>
SAS	<i>Substation Automation System</i>
SAT	<i>Site Acceptance Test</i>
SCD	<i>Substation Configuration Description</i>
SCL	<i>Substation Configuration Language</i>
SCSM	<i>Specific Communication Service Mapping</i>
SLAN	<i>Substation Local Area Network</i>

SNTP	<i>Simple Network Time Protocol</i>
SSD	<i>System Specification Description</i>
SV	<i>Sampled Values</i>
TC	Transformador de Corrente
TP	Transformador de Potencial
UAC	Unidade de Aquisição e Controle
UCA	<i>Utility Communication Architecture</i>
UML	<i>Unified Modeling Language</i>
UTR	Unidade Terminal Remota
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVO GERAL	15
1.2	OBJETIVOS ESPECÍFICOS	15
1.3	JUSTIFICATIVA	16
1.4	ESTRUTURA DO TRABALHO	17
2	TERMINOLOGIAS E CONCEITOS FUNDAMENTAIS	18
2.1	PROCESSAMENTO DIGITAL DE SINAIS	18
2.2	SISTEMAS DE TEMPO REAL	19
2.3	REDES DE COMUNICAÇÃO	19
2.4	DISPOSITIVOS ELETRÔNICOS INTELIGENTES	20
2.5	SISTEMAS DE AUTOMAÇÃO DE SUBESTAÇÕES	21
2.5.1	Situação Histórica e Atual	21
2.5.2	Situação Desejada	22
2.6	ORGÃOS INTERNACIONAIS DE PADRONIZAÇÃO E A IEC-61850	22
2.7	LINGUAGEM UML	23
3	NORMA IEC-61850	25
3.1	ESTRUTURA GERAL DA NORMA	25
3.2	GERENCIAMENTO E CICLO DE VIDA	28
3.3	REQUISITOS DE COMUNICAÇÃO E FUNÇÕES DE DISPOSITIVOS	29
3.4	LINGUAGEM DE CONFIGURAÇÃO DE SUBESTAÇÃO	30
3.5	MODELO DE DADOS	31

3.6	INTERFACE ABSTRATA DE SERVIÇOS DE COMUNICAÇÃO	33
3.7	PROTOCOLOS ESPECÍFICOS DE COMUNICAÇÃO	34
3.8	TESTES DE CONFORMIDADE	36
4	RESULTADOS E DISCUSSÃO	37
4.1	ASPECTOS NÃO ESPECIFICADOS NA IEC-61850	37
4.2	ESTRUTURA GERAL DA ARQUITETURA PROPOSTA	38
4.3	ESTRUTURA DE PROCESSOS E <i>THREADS</i> PARA A ARQUITETURA PROPOSTA	43
4.4	CLASSES ADICIONAIS E SEQUÊNCIAS DE EXECUÇÃO SUGERIDAS PARA A ARQUITETURA PROPOSTA	47
5	CONSIDERAÇÕES FINAIS	58
5.1	CONCLUSÕES	58
5.2	TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO	59
	REFERÊNCIAS	60

1 INTRODUÇÃO

Automação de subestação é o gerenciamento supervisor e controle dos sistemas de distribuição de energia elétrica (OZANSOY; ZAYEGH; KALAM, 2009a). Nas últimas décadas esses sistemas estão se tornando cada vez maiores, mais dinâmicos e distribuídos, aumentando substancialmente o interesse na automação de subestações, visto que um dos principais objetivos da automação de subestação é melhorar a operação, manutenção e eficiência da mesma (OZANSOY; ZAYEGH; KALAM, 2009b). Entretanto, a diversidade de soluções baseadas em *software* e protocolos proprietários de cada fabricante, muitas vezes incompatíveis entre si, apresentam dificuldades para a expansão e evolução desses sistemas. Nesse contexto, o uso de técnicas de comunicação efetivas, capazes de conectar os diversos dispositivos de controle, proteção e monitoramento presentes numa subestação é um fator crítico para o sucesso na automação de subestação.

A padronização torna-se então uma palavra chave para o desenvolvimento da conectividade e interoperabilidade entre sistemas de automação de subestação (OZANSOY; ZAYEGH; KALAM, 2009a). Na última década, muito trabalho foi feito no sentido de definir uma linguagem única de comunicação entre os diversos dispositivos utilizados na automação de subestações. Como resultado desse trabalho que envolveu organizações como o EPRI e a IEC, foi elaborada a norma IEC-61850, que define um conjunto de requisitos técnicos necessários para permitir que sistemas e equipamentos de diferentes fabricantes e gerações tecnológicas possam se comunicar por meio de uma rede de comunicação padronizada (IEC-61850-1, 2003).

1.1 OBJETIVO GERAL

Projetar e modelar, usando a linguagem UML, uma estrutura lógica de componentes capaz de representar a arquitetura base de referência de um sistema de automação de subestação aderente à norma IEC-61850 e que detalhe aspectos de implementação não especificados na normativa.

1.2 OBJETIVOS ESPECÍFICOS

- Caracterizar a infra-estrutura e os conceitos peculiares dos sistemas de automação de subestação;
- Analisar o padrão internacional IEC-61850;

- Identificar estruturas lógicas necessárias para implementar um sistema de automação e que não foram especificadas na normativa;
- Projetar e modelar, aplicando linguagem UML, componentes lógicos que definam a arquitetura base para implementação de um *software* de automação de subestação baseado na norma IEC-61850.

1.3 JUSTIFICATIVA

A norma IEC-61850 deverá causar um impacto significativo sobre como os sistemas de energia elétrica serão projetados e construídos durante os próximos anos, efetivamente reduzindo a diversidade e complexidade de soluções para automação nesse segmento, minimizando custos com operação, manutenção e projetos de engenharia (OZANSOY; ZAYEGH; KALAM, 2009b). A normativa utiliza uma abordagem dirigida a modelo para descrever a comunicação entre dispositivos em uma subestação, aplicando modelos de dados orientado a objeto para descrever propriedades e funcionalidades para serem implementadas e controladas, definindo também um conjunto de serviços para interação entre dispositivos. Entretanto, a norma foi modelada de maneira abstrata, definindo o que os modelos devem proporcionar e não como eles são construídos (OZANSOY; ZAYEGH; KALAM, 2009a).

As implicações no desenvolvimento e os detalhes de implementação, que estão relacionados ao “como” os modelos previstos na norma podem ser implementados para constituir um *software*, são as principais motivações para elaboração do presente trabalho. Um aspecto significativo para a construção de um *software* passível de evolução e que garanta interoperabilidade entre diferentes equipamentos e aplicações, atuais e futuras, aderentes a normativa, é a definição da arquitetura base do *software*. Esta arquitetura representa como será organizado a estrutura lógica do *software*, dos processos, componentes e suas dependências e relacionamentos. Nesse contexto, adota-se a linguagem UML como ferramenta para que a arquitetura base seja representada em uma linguagem livre de ambiguidades e amplamente utilizada na modelagem de *software*.

A definição de um modelo de arquitetura base para implementação de aplicações baseadas na IEC-61850 contribuirá para que futuros projetos de desenvolvimento de *software* aderentes a normativa tenham um referencial para desenvolver o projeto detalhado do *software*, partindo de um modelo arquitetural previamente definido e que permite aprimoramentos futuros, além de contribuir para a redução do tempo e esforço empregado à fase de engenharia de *software*.

1.4 ESTRUTURA DO TRABALHO

O presente trabalho está estruturado em capítulos e seções. O Capítulo 1 apresenta os objetivos e justificativa do trabalho e contextualiza o leitor no assunto abordado. Terminologias e conceitos fundamentais relacionados com o assunto abordado e necessários para compreender o restante do trabalho são apresentados no Capítulo 2, onde também é descrito sobre o contexto histórico e atual dos sistemas de automação de subestações e do papel que organizações internacionais de padronização tem desempenhado nesse contexto.

O Capítulo 3 apresenta um estudo da estrutura, características e requisitos da norma IEC-61850, aplicada à redes de comunicação e sistemas em subestações, descrevendo em seções específicas sobre o conteúdo abordado em cada um dos principais documentos que compõem a norma.

Com base no estudo da norma e de suas características e modelos conceituais, são identificados no Capítulo 4 aspectos e detalhes não contemplados na normativa. Posteriormente, aplicando conhecimentos de modelagem e engenharia de software, são apresentados e discutidos os diagramas UML elaborados neste trabalho para definir o modelo de arquitetura base de aplicações baseadas na norma IEC-61850. Por fim, no Capítulo 5, são apresentadas as considerações finais e sugestões de trabalhos futuros.

2 TERMINOLOGIAS E CONCEITOS FUNDAMENTAIS

Visando o melhor aproveitamento e compreensão dos textos apresentados ao longo deste trabalho, são abordados a seguir alguns conceitos técnicos fundamentais e terminologias relacionadas ao tema em questão, sendo também apresentada uma síntese da situação histórica, da evolução de tecnologias e de padrões pertinentes ao tema abordado neste trabalho.

2.1 PROCESSAMENTO DIGITAL DE SINAIS

O Processamento Digital de Sinais (DSP - *Digital Signal Processing*) é utilizado em qualquer área onde informações são manipuladas ou controladas por um processador digital. O processamento digital de sinais oferece muitas vantagens e flexibilidade em relação ao tratamento analógico. Diversos procedimentos que só poderiam ser obtidos em sistemas analógicos com a utilização de equipamento especializado, complexo e caro, podem ser mais eficientemente executados no domínio digital (SCHWANKE, 2000).

A maioria dos sinais encontrados diretamente em ciência e engenharia são contínuos (analógicos): intensidade luminosa que se modifica com a distância, tensão que varia no tempo, velocidade de uma reação química dependente da temperatura, etc. A Conversão Analógica/Digital (ADC) e a Conversão Digital/Analógico (DAC) são os processos que possibilitam aos computadores digitais interajam com estes sinais (SMITH, 1998).

As formas mais comuns de DSP têm como variáveis de entrada os sinais analógicos, os quais são amostrados em intervalos regulares no tempo e convertidos para sua forma digital. A Figura 1 ilustra o processo de conversão analógica/digital.

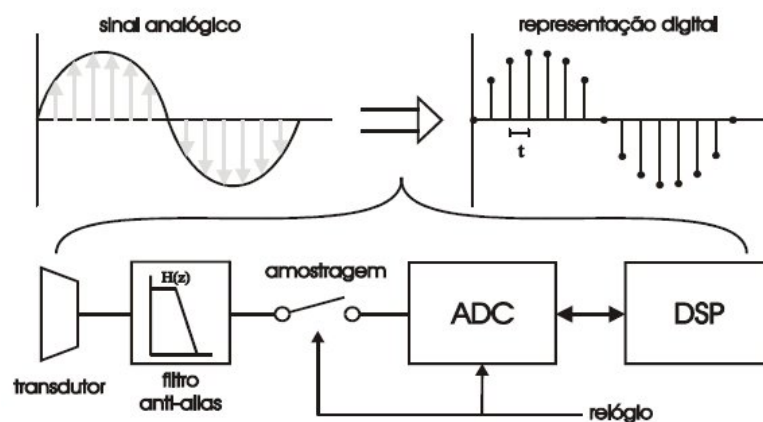


Figura 1: Conversão Analógica/Digital
Fonte: SCHWANKE (2000).

Em certas situações, a aquisição e processamento de sinais devem obedecer as restrições

temporais características dos sistemas de tempo real.

2.2 SISTEMAS DE TEMPO REAL

Um sistema computacional de tempo real é um sistema cuja correção depende não somente do resultado lógico dos processamentos, mas também do instante em que os resultados são produzidos (KOPETZ, 1997). Os sistemas de tempo real, portanto, devem ter a habilidade de executar um comando ou instrução e disponibilizarem a resposta em um tempo relativamente previsível, ou seja, que seja possível estimar o tempo de processamento necessário (LOCKE, 2000).

As aplicações de tempo real diferenciam-se em grau de criticidade. De um modo simplificado, algumas possuem restrições temporais rígidas (*hard real-time*) e outras possuem restrições temporais menos rígidas (*soft real-time*). Entretanto, na prática existem diversas classificações utilizadas.

Para a implementação de sistemas de tempo real, onde as noções de tempo e de concorrência são tratadas explicitamente, conceitos e técnicas de escalonamento representam um ponto central para garantir que as tarefas sejam executadas dentro de seus *deadlines*, isto é, do prazo máximo desejado para a conclusão de uma tarefa. É de fundamental importância o conhecimento dos períodos de ativação e a escolha de uma abordagem de escalonamento adequada à classe de problemas que o sistema deve tratar (OLIVEIRA; FARINES; FRAGA, 2000). Neste contexto, os sistemas operacionais de tempo real precisam fornecer serviços previsíveis para as tarefas de aplicações, de maneira que as restrições temporais possam ser analisadas estaticamente possibilitando ter seu atendimento garantido em tempo de execução (KOPETZ, 1997).

2.3 REDES DE COMUNICAÇÃO

As redes de comunicação surgiram da necessidade de interconexão entre diversos computadores para que pudessem trocar informações. “As redes de computadores estabelecem a forma-padrão de interligar computadores para o compartilhamento de recursos físicos ou lógicos” (MENDES, 2007).

Em poucas décadas os computadores foram diminuindo de tamanho e se espalhando pelos mais diversos meios, gerando necessidade de avanços tecnológicos na comunicação e transmissão de dados entre os mais diversos dispositivos. Para suportar a demanda crescente de comunicação entre equipamentos distintos, surgiram diversas tecnologias e padrões de comunicação, porém a tecnologia *Ethernet* é atualmente a mais popular.

A concepção da *Ethernet* data de 22 de maio de 1973, quando Bob Metcalfe, que trabalhava na Xerox, Califórnia, escreveu um memorando descrevendo a rede *Ethernet* que ele havia inventando para intercomunicação avançada entre estações de trabalho (SPURGEON, 2000). Mais tarde o *Institute of Electrical and Electronics Engineers* (IEEE) padronizou as redes *Ethernet* sob número 802.3 (*Carrier Sense Multiple Access with Collision Detection*). Atualmente o padrão continua evoluindo e suportando redes com tráfego cada vez maior.

Além da tecnologia de rede, a interconexão entre computadores também depende de protocolos de comunicação, os quais definem um conjunto de regras que coordenam e asseguram o transporte de informações. Visando identificar as tarefas fundamentais que devem ser implementadas para a comunicação entre computadores, foi desenvolvido pela ISO (*International Organization for Standardization*) o Modelo de Referência OSI (*Open Systems Interconnection Reference Model* ou OSI-RM). O modelo divide o tratamento de informações de redes de computadores em sete níveis, de maneira a se obter camadas de abstração. Cada protocolo implementa uma funcionalidade assinalada a uma determinada camada (MODELO... , 2011).

2.4 DISPOSITIVOS ELETRÔNICOS INTELIGENTES

As características dos Sistemas de Automação de Subestações (SAS) têm evoluído sensivelmente com a utilização de dispositivos de proteção baseados em processadores digitais. Esses dispositivos têm apresentado um caráter multifuncional, relacionando, além das funções de proteção, muitas funções adicionais, tais como, medida, registro de eventos, controle, monitoração de qualidade de energia, entre outras, sendo caracterizados como uma evolução dos relés de proteção e denominados Dispositivos Eletrônicos Inteligentes (IED - *Intelligent Electronic Devices*).

Uma das características dos IEDs é permitir a execução de funções de proteção e controle distribuídas sobre uma rede local de subestação (SLAN). A utilização de uma rede local permite a substituição da fiação de comando de cobre rígida por uma instalação de comando digital, bem como o aperfeiçoamento do controle lógico com melhoria na funcionalidade de proteção. Assim esses dispositivos começam a ganhar uma larga aceitação, sendo reconhecidos como essenciais para a operação eficiente e gerenciamento de uma subestação moderna (PAULINO, 2007).

Os IEDs tem sido cada vez mais utilizados nas subestações elétricas à medida que agregam mais recursos. O uso dos IEDs permite uma redução no custo de implantação, de manutenção, no número de cabos e equipamentos necessários à sua utilização, possibilitando troca de informações mais rápidas, simplificação do projeto, maior confiabilidade, além de permitir a sincronização temporal dos dispositivos (LACERDA; CARNEIRO, 2010).

2.5 SISTEMAS DE AUTOMAÇÃO DE SUBESTAÇÕES

Os Sistemas de Automação de Subestações (SAS) têm como principais objetivos a proteção, controle e supervisão de sistemas elétricos de potência (JOSÉ, 2010). A intensificação da automação vem implicando em profundos impactos nos processos produtivos em diversos segmentos industriais e em toda a sua cadeia produtiva. Nas subestações de energia que alimentam esses processos produtivos não é diferente, onde equipamentos de medição e proteção eletromecânicos obsoletos estão sendo substituídos por equipamentos mais modernos baseados em processadores digitais (SOUTO; FONSECA, 2007).

2.5.1 Situação Histórica e Atual

Historicamente os primeiros relés de proteção utilizados eram de tecnologia eletromecânica e os sistemas de controle e supervisão associados, por sua vez, eram constituídos por chaves de controle e chaves seletoras, instrumentos de medição indicativa e anunciadores de alarme. Os dispositivos de controle eram distribuídos em painéis de controle e os relés em painéis de proteção. Nas subestações e usinas de maior porte, um grande espaço era necessário para acomodar todos estes painéis e ligações físicas realizadas através de fios, que requeria um conjunto independente de contatos para cada uma das diferentes lógicas necessárias, chegando, muitas vezes, a dezenas ou até centenas de quilômetros de cabos de controle (PEREIRA, 2005).

Alguns dos primeiros equipamentos digitais instalados em subestações e usinas na área de supervisão e controle foram as Unidades Terminais Remotas (UTR). Inicialmente, estas unidades eram equipamentos de aquisição de dados e execução de comandos. Posteriormente, as UTRs passaram a ser dotadas de inteligência própria, sendo capazes de executar diversas funções extras como autoteste, validação de medidas e estado de equipamentos, registro do instante de ocorrências e armazenamento de alarmes e eventos. Com essas novas funcionalidades, e pelo fato de possuírem processamento próprio, as UTRs, em algumas empresas, passaram a ser denominadas Unidades de Aquisição e Controle (UAC). Alguns equipamentos digitais, tais como o registrador sequencial de eventos, o oscilógrafo digital e equipamentos de medição digital se desenvolveram de forma independente. Estes equipamentos, juntamente com as UACs e demais equipamento dotados de processamento próprio, foram identificados de uma forma genérica como IEDs (*Intelligent Electronic Devices*) (PEREIRA, 2005).

Após mais de 20 anos de desenvolvimento, a tecnologia digital aplicada a sistemas de proteção, supervisão e controle atingiu um estágio de maturidade. Produtos globalizados são projetados para o mercado mundial e atendem à requisitos relevantes de diversas normas interna-

cionais. Entretanto, atualmente é comum encontrarmos em usinas e subestações uma variedade de tecnologias modernas e legadas, com carência de gerenciamento remoto e de padronização de protocolos e interfaces entre diferentes módulos e componentes. Os sistemas de automação de subestações estão evoluindo de arquiteturas centralizadas, que utilizavam um extenso emaranhado de cabos metálicos, para soluções distribuídas, baseadas em redes de comunicação de dados (JOSÉ, 2010).

2.5.2 Situação Desejada

É inevitável a coexistência, em uma mesma subestação ou usina, de dispositivos fornecidos por diferentes fabricantes, inclusive dispositivos pertencentes a gerações tecnológicas distintas. Estes dispositivos e seus respectivos sistemas precisam se comunicar entre si, em tempo real, com garantia e confiabilidade. A situação ideal é aquela onde, independentemente de fornecedor ou geração tecnológica, tais dispositivos e sistemas apresentam plena capacidade de interoperabilidade, com a mínima complexidade de implantação, manutenção e evolução dessa estrutura.

Devido ao excesso de protocolos existentes no mercado e a necessidade de reduzir custos operacionais do sistema elétrico, tornou-se imprescindível a criação de uma norma que possibilitasse a redução da quantidade de protocolos, que suportasse expansão em longo prazo, que fosse adaptável à rápida evolução das tecnologias e que ainda permitisse aumentar a confiabilidade do sistema elétrico. Nesse contexto os órgãos internacionais de padronização exerceram um papel fundamental no sentido de padronizar e normatizar as interfaces para SASs e IEDs.

2.6 ORGÃOS INTERNACIONAIS DE PADRONIZAÇÃO E A IEC-61850

No começo dos anos 90 a arquitetura UCA (*Utility Communications Architecture*) começou a ser desenvolvida nos Estados Unidos, no EPRI (*Electric Power Research Institute*), com o objetivo de desenvolver uma estrutura de comunicação em tempo real comum a todas as empresas do setor de energia elétrica (GURJÃO; CARMO; SOUZA, 2006). Com base nestes estudos a versão 2.0 da UCA foi publicada pelo IEEE (*Institute of Electrical and Electronics Engineers*) como relatório técnico TR1550 de 1999. Paralelamente, a IEC (*International Electrotechnical Commission*) tentava normalizar as interfaces para dispositivos de telecontrole, por meio da série de normas IEC-60870-5, através dos comitês técnicos IEC-TC57 e IEC-TC95 (MIRANDA, 2009).

Em 1994, reconhecendo a necessidade de uma padronização mais geral cobrindo redes de

comunicação e sistemas em subestações, foram criados três grupos de trabalho ligados ao comitê técnico IEC-TC57, reunindo especialistas de vários países com experiência nos protocolos segundo as normas IEC-60870-5 e UCA 2.0. A IEC e EPRI decidiram então unir esforços para obter um padrão internacionalmente aceito. Esse padrão é conhecido como IEC 61850 - Redes de Comunicação e Sistemas em Subestações (do inglês, IEC 61850 - *Communication Networks and Systems in Substation*) (MIRANDA, 2009).

Apesar de ainda estar em desenvolvimento, a norma IEC-61850 define fundamentalmente o modelo de dados e a pilha de protocolos para a comunicação entre equipamentos, tendo despertado interesse das empresas, pois pode solucionar o problema da comunicação entre dispositivos de diferentes fabricantes e gerações tecnológicas (GURJÃO; CARMO; SOUZA, 2006).

2.7 LINGUAGEM UML

Alguns dos modelos definidos na norma IEC-61850 são documentados usando diagramas da linguagem UML (*Unified Modeling Language*), empregando conceitos e técnicas de orientação a objeto para descrever, de maneira abstrata, propriedades e funcionalidades a serem implementadas e controladas, definindo também um conjunto de serviços para interação entre dispositivos.

A UML é uma linguagem de modelagem, independente de processos e tecnologias de implementação, utilizada para especificação, visualização, construção e documentação de artefatos de sistemas de software, podendo, por meio de notações e regras semânticas, representar tanto os aspectos conceituais quanto lógicos (OMG, 2005). Entretanto, a linguagem se limita a representar um sistema através de um conjunto de diagramas, onde cada diagrama se refere a uma visão parcial do sistema, que em conjunto representam o todo integrado e coerente.

A linguagem UML está em contínua evolução, sendo que a versão 2.0 propõe 13 diagramas diferentes para representar os aspectos estruturais, comportamentais e físicos de um *software* (BOOCH; RUMBAUGH; JACOBSON, 2005). Dentro do escopo do presente trabalho serão utilizados os seguintes diagramas:

- **Diagrama de Classe:** Este diagrama é uma representação da estrutura e relacionamentos das classes que descrevem o modelo de um sistema. É um diagrama que descreve o que interage com o que, porém não descreve quando e como isso ocorre;
- **Diagrama de Sequência:** Este diagrama detalha os relacionamentos e interações que ocorrem entre as classes de um sistema. O diagrama mostra como e quando as interações

acontecem, que mensagens são enviadas e recebidas, organizado de acordo com o tempo no qual cada operação acontece;

- **Diagrama de Componente:** Este diagrama ilustra como as classes deverão ser organizadas de maneira a compor módulos ou componentes de um sistema, aumentando o nível de abstração e definindo funções específicas para cada componente, facilitando a reutilização;
- **Diagrama de Pacote:** Este diagrama representa partes de um sistema dividido em agrupamentos lógicos de elementos relacionados. Um pacote representa um grupo de classes, ou outros elementos, que possuem alguma característica ou finalidade em comum;
- **Diagrama de Implantação (*Deployment*):** Este diagrama representa a configuração e arquitetura física de *software* e *hardware*, demonstrando o relacionamento entre os vários componentes envolvidos.

3 NORMA IEC-61850

Um dos principais objetivos da norma internacional IEC-61850 é garantir a interoperabilidade entre IEDs de diferentes fabricantes, permitindo o uso e a troca de informações a fim de assegurar a correta operação e cooperação entre esses equipamentos, além de suportar desenvolvimentos tecnológicos futuros sem requerer alterações significativas no *software* e *hardware* dos Sistemas de Automação de Subestações. Esta necessidade surge basicamente da dificuldade encontrada nos processos de integração de informações durante as diferentes etapas de implementação na automação de subestações, principalmente quando distintos equipamentos ou componentes, frequentemente de diferentes fornecedores, devem ser integrados. A Norma IEC-61850 surge então como um requisito de mercado derivado de experiências práticas, evoluções tecnológicas, especificações de clientes e de métodos de engenharia disponibilizados por diferentes fabricantes (SANTOS; PEREIRA, 2007).

Outro objetivo da norma IEC-61850 é possibilitar a comunicação entre IEDs com alta velocidade e confiabilidade, por meio redes de comunicação, permitindo a substituição dos cabos de controle e reduzindo custos. A IEC-61850, ao normatizar os protocolos de comunicação dos IEDs, possibilita a eliminação de conversores de protocolos, reduzindo custos de implantação e de treinamento do corpo técnico para operar os diversos tipos de dispositivos de diferentes fabricantes. Desta maneira evitando atrasos, reduzindo possíveis erros e assim aumentando a confiabilidade do sistema.

As características do padrão IEC-61850 permitem que o SAS seja considerado uma plataforma aberta de proteção e automação de subestações, independentemente de fornecedor. Os sinais do processo e de outros IEDs são configurados em linguagem SCL (*Substation Configuration Language*) e os modelos de funções e objetos estão padronizados, otimizando a reutilização e consistência de *software* aplicativos. Para alcançar estes objetivos a norma utiliza uma abordagem orientada a objeto.

3.1 ESTRUTURA GERAL DA NORMA

A norma IEC-61850 é composta por uma série de documentos, estruturada em partes conforme ilustrado na Tabela 1. Cada parte da norma descreve sobre características específicas, sendo que ao longo do texto a norma ainda referencia diversas outras normas e padrões já existentes, o que representa uma característica interessante pois reduz a extensão da mesma e ainda permite que seja reutilizado implementações já existentes.

Podemos observar na Tabela 1 que as primeiras partes da norma descrevem sobre os aspec-

Tabela 1: Série de Documentos que definem a Norma IEC-61850

Característica	Parte	Descrição	Publicação
Aspectos do Sistema	1	Introdução e Visão Geral	04/2003
	2	Glossário	01/2002
	3	Requisitos Gerais	01/2002
	4	Gerenciamento de Sistema e Projeto	01/2002
	5	Requisitos de Comunicação para Funções e Modelos de Dispositivos	07/2003
Configuração	6	Linguagem de Configuração para IEDs de Subestações Elétricas (SCL)	03/2004
Estrutura de Comunicação Básica para Equipamentos de Subestações e Alimentadores	7.1	Princípios e Modelos	07/2003
	7.2	Interface Abstrata de Serviços de Comunicação (ACSI)	05/2003
	7.3	Classe de Dados Comum (CDC)	05/2003
	7.4	Classes de Nós Lógicos e de Dados Compatíveis	05/2003
Mapeamento de Serviços de Comunicação Específicos	8.1	Mapeamento para MMS (ISO/IEC 9506 Parte 1 e Parte 2) e para ISO/IEC 8802-3	05/2004
	9.1	Valores Amostrais sobre Enlace Serial Unidirecional <i>Multidrop</i> Ponto-a-Ponto	05/2003
	9.2	Valores Amostrais sobre ISO/IEC 8802-3	04/2004
Ensaaios	10	Testes de Conformidade	06/2005

Fonte: MIRANDA (2009).

tos gerais, como por exemplo, a contextualização inicial; as terminologias específicas e definições utilizadas na conjuntura dos sistemas de automação de subestações e nas várias partes da norma; os aspectos sobre gerenciamento de projeto e sistema, como qualidade, manutenibilidade, atualização, garantias e condições ambientais. A parte seguinte menciona sobre o processo e linguagem necessária para configuração da subestação e outras informações relacionadas com a parametrização do sistema, e que podem ser interpretadas por qualquer ferramenta compatível com o padrão IEC-61850. Na sequência as partes da norma abordam os princípios, atributos e modelos abstratos que permitem alcançar o objetivo principal da norma, a interoperabilidade entre equipamentos de fabricantes e gerações tecnológicas distintas. Posteriormente, o foco concentra-se na implementação real de protocolos de comunicação específicos, baseados em padrões abertos e na tecnologia de rede *Ethernet*. Por fim, e não menos importante, é dedicado uma parte da norma para descrever sobre metodologias e arquiteturas de testes de conformidade, de desempenho, de qualidade e critérios de aceitação.

As diversas funções de um sistema de automação de subestação, como por exemplo, proteção, medição, monitoração e controle, são subdivididas em objetos denominados nós lógicos, os quais podem se comunicar entre si para desempenhar determinada função. Cada nó lógico possui seu próprio conjunto de dados, denominado classes ou objetos de dados. Os dados são compartilhados entre os nós lógicos segundo regras que são chamadas serviços. O conjunto de dados e serviços é mapeado, constituindo uma especificação de mensagens. Entre as mensagens, algumas apresentam restrições temporais, devendo ser transferidas respeitando limites de tempo antecipadamente previstos, sob qualquer condição de operação do sistema. As aplicações e a transmissão de mensagens através de pacote de dados são funções separadas e independentes, permitindo que a tecnologia da comunicação sofra evoluções sem que haja necessidade de alterar a estrutura de dados das aplicações e vice versa (PEREIRA, 2005). A Figura 2 ilustra um conjunto de nós lógicos e a estrutura lógica de comunicação entre os mesmos.

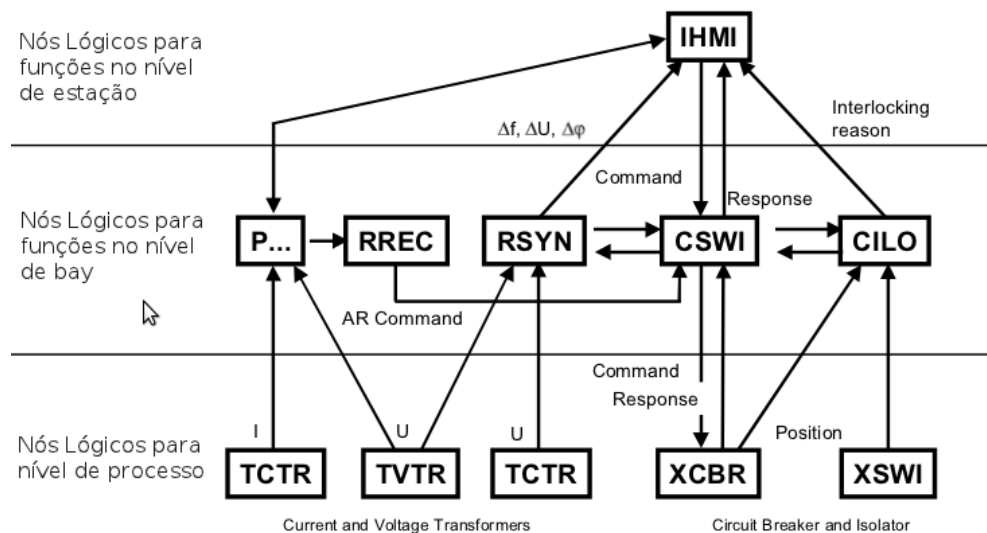


Figura 2: Esquema de Comunicação entre Nós Lógicos
Fonte: Adaptado de IEC-61850-5 (2003).

Normalmente um SAS aderente a norma está estruturado com dois barramentos: barramento de processo e barramento de estação; e três níveis hierárquicos: nível de estação, nível de bay ou vão, e nível de processo. No nível de processo estão as conexões com os equipamentos primários da subestação, como por exemplo, TCs (Transformadores de Corrente), TPs (Transformadores de Potencial), disjuntores e chaves seccionadoras. No nível de bay estão os equipamentos de proteção e controle. Já no nível de estação a principal finalidade é a supervisão e controle da subestação através de uma Interface Homem Máquina (IHM), permitindo também a conexão com um outro possível nível hierárquico ainda superior (MIRANDA, 2009). A comunicação entre nível de processo e bay ocorre pelo barramento de processo enquanto que a comunicação entre nível de estação e bay ocorre pelo barramento de estação. Entretanto,

é possível que todas as interfaces lógicas de comunicação utilizem um único barramento, se isto satisfizer os requisitos de desempenho (IEC-61850-1, 2003). A Figura 3 apresenta uma ilustração da estrutura de barramentos e níveis hierárquicos.

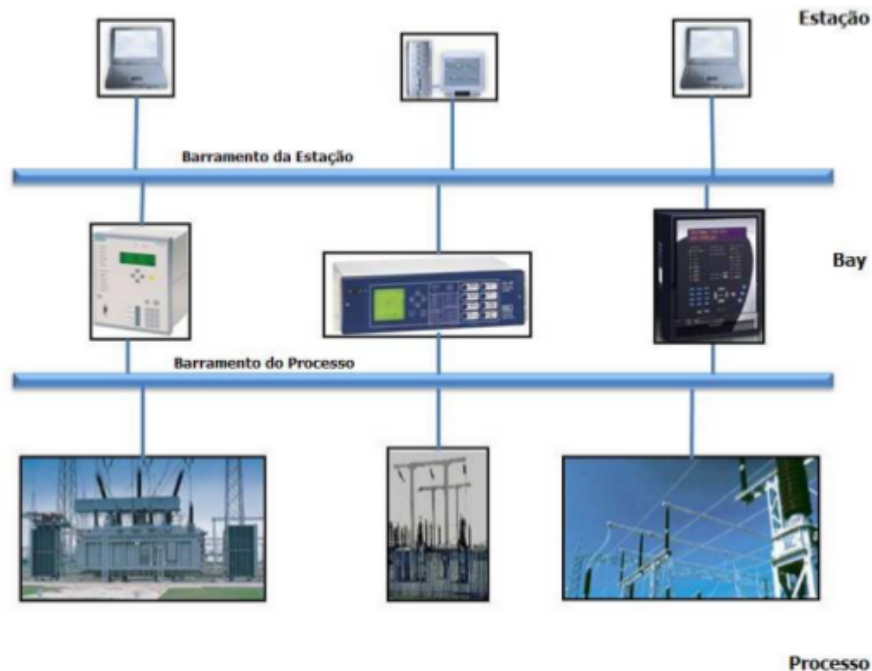


Figura 3: Barramentos de Comunicação e Níveis Hierárquicos
Fonte: JUNIOR et al. (2010).

3.2 GERENCIAMENTO E CICLO DE VIDA

A norma descreve algumas exigências básicas relativas ao processo de gerenciamento de projeto e sistema para automação de subestação, além de algumas características necessárias para ferramentas de teste e engenharia, com respeito aos seguintes tópicos (IEC-61850-4, 2002):

- Processos de engenharia e ferramentas de suporte;
- Ciclo de vida de todo o sistema e seus IEDs;
- Garantia de qualidade iniciada no estágio de desenvolvimento e encerrada com a descontinuidade e desativação do SAS e seus IEDs.

Os processos de engenharia e as ferramentas utilizadas criam condições para adaptar um SAS às especificações da subestação e sua filosofia de operação. A fase de engenharia inclui a definição das configurações de *hardware* necessárias para a subestação, a definição dos IEDs e suas interfaces com outros IEDs e com o ambiente, o dimensionamento das funcionalidades e

quantidade de sinais envolvidos, e a parametrização e documentação do projeto (MIRANDA, 2009).

O ciclo de vida de um SAS e seus IEDs está sujeito a diferentes pontos de vistas: do fabricante e do cliente (IEC-61850-4, 2002):

- Para o fabricante o ciclo de vida inclui o período entre o início de produção até a descontinuidade da família de produtos do SAS;
- Para o cliente o ciclo de vida compreende o período entre o comissionamento da primeira instalação do SAS com base em uma família de produtos até a desativação da última instalação.

A norma IEC-61850 define que o fabricante deve anunciar a descontinuidade de um produto e prestar suporte após a interrupção da produção. Já a qualidade é entendida como uma tarefa comum às duas entidades, fabricante e cliente (IEC-61850-4, 2002). Por exemplo, o fabricante deve estabelecer e manter um sistema de qualidade de acordo com a ISO-9001, enquanto que o cliente é responsável por assegurar que as condições de operação e ambiente satisfazem as condições estabelecidas na documentação técnica do SAS e seus produtos.

3.3 REQUISITOS DE COMUNICAÇÃO E FUNÇÕES DE DISPOSITIVOS

A parte 5 da norma IEC-61850 define os requisitos para comunicação das funções implementadas nos diversos níveis hierárquicos do sistema de automação de subestação. As funções são estabelecidas e determinadas de acordo com as tarefas a serem executadas na subestação, como monitoramento, controle e proteção. A norma padroniza os dados de configuração, entrada e saída dessas funções (PAULINO; SIQUEIRA; CARMO, 2010).

Conforme abordagem apresentada na Figura 4, as Funções (*F - Functions*) são decompostas em Nós Lógicos (*LN - Logical Nodes*) que podem residir em um ou mais Dispositivos Físicos (*PD - Physical Devices*). Os nós lógicos podem trocar informações através de Conexões Lógicas (*LC - Logical Connections*), sendo que qualquer nó lógico é parte de um dispositivo físico e qualquer conexão lógica é parte de uma Conexão Física (*PC - Physical Connection*) (IEC-61850-5, 2003).

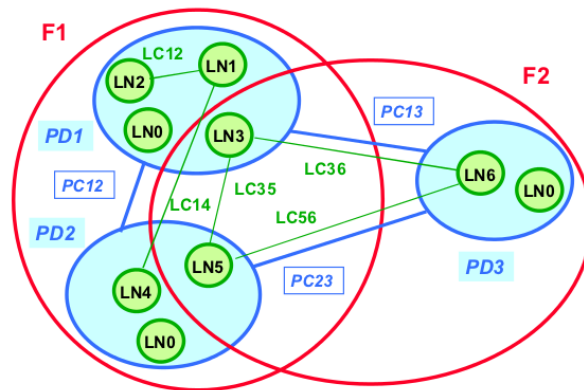


Figura 4: Estrutura Lógica de Comunicação
Fonte: IEC-61850-5 (2003).

3.4 LINGUAGEM DE CONFIGURAÇÃO DE SUBESTAÇÃO

Frente às possibilidades de uso dos nós lógicos em diferentes dispositivos físicos e a necessidade de determinar e configurar os caminhos e parâmetros de troca de informação entre esses nós lógicos, a norma especifica uma linguagem formal de descrição da configuração para sistemas de automação de subestações, chamada de Linguagem de Configuração de Subestação (SCL - *Substation Configuration Language*). Um dos principais objetivos da linguagem SCL é a uniformização da nomenclatura utilizada através de um modelo único de descrição de dados, criando um vocabulário comum. A Figura 5 apresenta uma representação UML simplificada desse modelo.

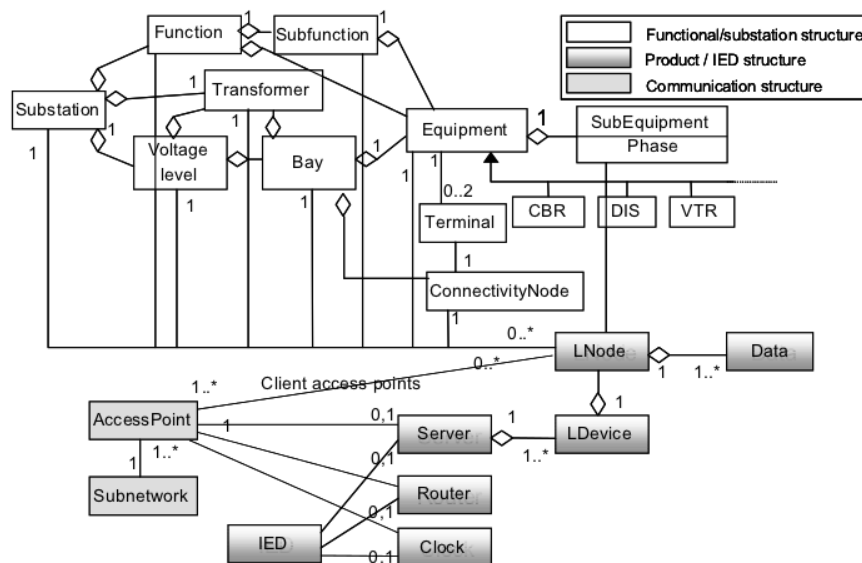


Figura 5: Representação UML da Linguagem SCL
Fonte: IEC-61850-6 (2004).

A definição de um modelo único permite a troca de informações entre as ferramentas de

engenharia dos IEDs e as ferramentas de engenharia do SAS. Essas ferramentas são aplicações de parametrização dos dados do sistema e ajustes dos IEDs usados na implantação do SAS. A troca de informações de maneira compatível entre ferramentas, inclusive de diferentes fabricantes, é possível devido a elaboração de arquivos comuns a todos esses fabricantes (PAULINO; SIQUEIRA; CARMO, 2010). A Figura 6 ilustra a estrutura da SCL e a utilização desses arquivos, conforme definido na IEC-61850.

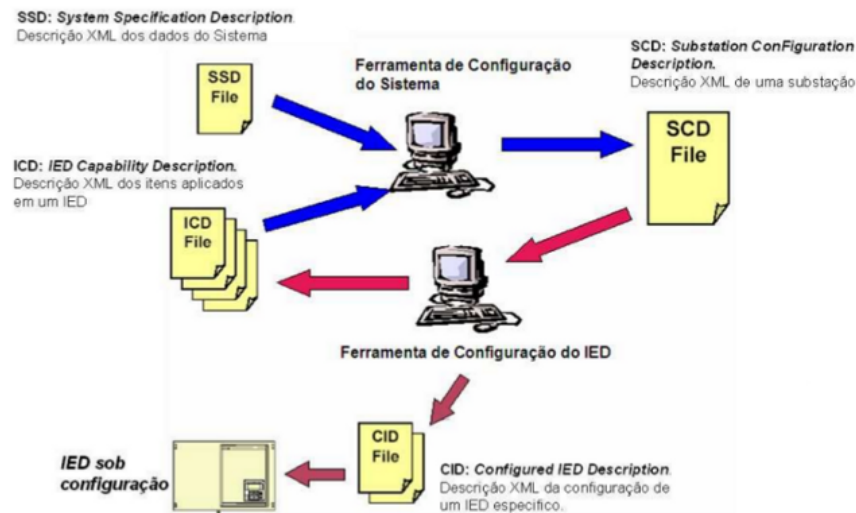


Figura 6: Estrutura de Arquivos da Linguagem SCL
Fonte: PAULINO, SIQUEIRA e CARMO (2010).

Todos os arquivos são formatados em XML (*Extensible Markup Language*) definido pela norma IEC-61850, e possuem as seguintes finalidades apresentadas na Tabela 2.

Tabela 2: Finalidade dos Arquivos da Linguagem SCL

Arquivo	Finalidade
SSD: <i>System Specification Description</i>	Descreve o diagrama e a funcionalidade da automação associado aos nós lógicos.
SCD: <i>Substation Configuration Description</i>	Descreve a configuração da subestação incluindo a rede de comunicação e informações sobre o fluxo de dados de comunicação.
ICD: <i>IED Capability Description</i>	Descreve as capacidades e pré-configurações dos IEDs. Neste arquivo estão descritas todas as funções suportadas por um IED.
CID: <i>Configured IED Description</i>	Descrição da configuração de um IED específico. Neste arquivo estão descritas as funções parametrizadas ou habilitadas pelo usuário para aquele IED.

Fonte: PAULINO, SIQUEIRA e CARMO (2010).

3.5 MODELO DE DADOS

Para a norma IEC-61850 a menor parte de uma função lógica dentro do sistema de automação é chamada de nó lógico. Esses nós lógicos são um agrupamento de dados e serviços

modelados segundo o paradigma de orientação a objetos, sendo que um equipamento físico pode hospedar diferente arranjos de nós lógicos.

Um IED é formado por um *hardware* que se conecta ao sistema de comunicação de dados através de um endereço de rede, e por um conjunto de funções estruturadas em nós lógicos, que caracterizam seu comportamento dentro do SAS (MIRANDA, 2009). Os nós lógicos mantêm sob controle uma relação de dados e atributos referentes a sua função em particular. A Figura 7 ilustra o modelo de dados de um equipamento.

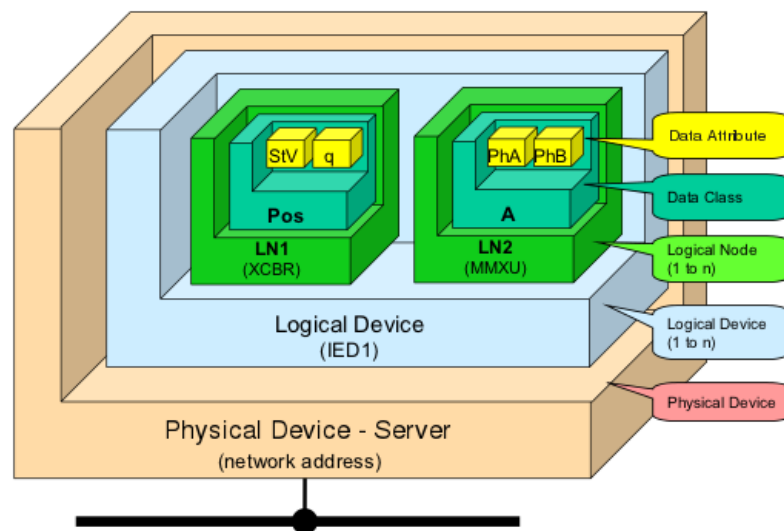


Figura 7: Representação Gráfica do Modelo de Dados de um IED
Fonte: Adaptado de PROUDFOOT (2002).

Para possibilitar a interoperabilidade na comunicação, o nome dos dados e atributos, as classes de dados comuns, os tipos de dados e outros elementos são especificados pela norma. Cada elemento de dado de um nó lógico está em conformidade com a especificação de uma classe de dados. Inicialmente a norma classificou os nós lógicos em 13 grupos, definindo 92 nós lógicos e 355 classes de dados (PAULINO; SIQUEIRA; CARMO, 2010). Novas versões da norma tendem a aumentar esses números.

Os nós lógicos definidos cobrem as aplicações mais comuns de subestações e equipamentos alimentadores, sendo que o foco principal foi a definição de modelos para os grupos de proteção e aplicações relacionadas a proteção, cobrindo 38 nós lógicos, isto porque funções de proteção, historicamente, apresentam grande importância para a segurança e confiabilidade da operação do sistema elétrico (IEC-61850-7-1, 2003). A Tabela 3 lista todos os grupos de nós lógicos definidos na IEC-61850, parte 7-4.

Tabela 3: Grupos de Nós Lógicos

Identificador do Grupo	Nome do Grupo	Quantidade de Nós Lógicos
A	Controle Automático	4
C	Controle Supervisório	5
G	Funções Genéricas	3
I	Interfaces e Arquivamento	4
L	Nós Lógicos do Sistema	3
M	Contador e Medição	8
P	Funções de Proteção	28
R	Funções Relacionadas a Proteção	10
S	Sensores e Monitoramento	4
T	Transformador de Instrumento	2
X	Disjuntor e Chave Seccionadora	2
Y	Transformador de Potência	4
Z	Equipamento do Sistema de Energia	15

Fonte: Adaptado de JOSÉ (2010).

3.6 INTERFACE ABSTRATA DE SERVIÇOS DE COMUNICAÇÃO

A estrutura de comunicação da IEC-61850 utiliza o conceito de “definição abstrata” para dados e serviços, permitindo mapeá-los para diferentes protocolos que atendam os dados e serviços requeridos. As partes 7-2 e 7-4 da norma definem, respectivamente, os serviços abstratos e os objetos de dados abstratos.

A Interface Abstrata de Serviços de Comunicação (ACSI - *Abstract Communication Service Interface*) é definida pela norma como um modelo de classe hierárquica de todos os dados que podem ser acessados e trocados, implantando a cooperação entre os vários dispositivos do sistema. Conforme ilustrado na Figura 8, o ACSI dispõe das seguintes interfaces abstratas (JOSÉ, 2010):

- Interface abstrata que descreve a comunicação entre um IED cliente e um IED servidor;
- Interface de comunicação para sistemas cujo tempo na troca de dados seja um fator crítico, como por exemplo, a troca de dados entre IEDs utilizados para proteção do sistema elétrico.

As referências a dados são feitas segundo o modelo de orientação a objetos, mantendo uma hierarquia iniciada pelo dispositivo lógico em nível superior até alcançar o nível inferior de atributo de dados, entretanto, o efetivo acesso e troca de informações ocorre por meio de uma camada de serviços abstratos de comunicação. A seguinte sintaxe pode referenciar de maneira genérica um determinado dado de um nó lógico (IEC-61850-7-2, 2003):

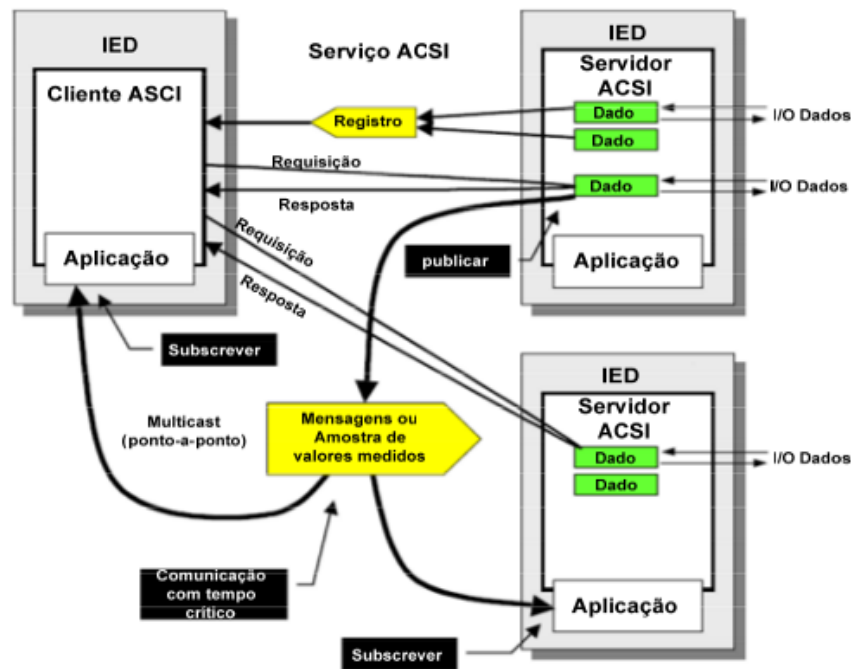


Figura 8: Estrutura de Comunicação ACSI
Fonte: MIRANDA (2009).

“DispositivoLógico/NóLógico.ObjetoDeDados.AtributoDeDados”

3.7 PROTOCOLOS ESPECÍFICOS DE COMUNICAÇÃO

Para que os serviços definidos no ACSI se tornem parte do mundo real estes devem ser mapeados sobre um protocolo “real”. Este mapeamento é chamado de SCSM (*Specific Communication Service Mapping*). De acordo com a norma, os serviços são mapeados para diferentes perfis de comunicação (IEC-61850-8-1, 2004):

- **Cliente/Servidor:** Núcleo de serviços ACSI orientado a conexão, baseado no protocolo MMS (*Manufacturing Message Specification*);
- **Time Sync:** Sincronismo de tempo utilizando o protocolo SNTP (*Simple Network Time Protocol*);
- **SV (Sampled Values):** Valores analógicos amostrados, como por exemplo, corrente e tensão;
- **GOOSE (Generic Object Oriented Substation Event):** Valores de estado e controle, orientado a eventos, transmitidos por *data set* (conjunto de dados) configuráveis;
- **GSSE (Generic Substation Status Event):** Valores de estado, orientado a eventos, transmitido em uma estrutura fixa.

Visando atender a diferentes requisitos de comunicação das subestações, as mensagens são classificadas conforme o desempenho (PAULINO; SIQUEIRA; CARMO, 2010). A Tabela 4 descreve os tipos de mensagens, classes de desempenho e respectivos perfis de comunicação vinculados.

Tabela 4: Tipos de Mensagens e Classes de Desempenho

Tipo	Classe de Desempenho	Perfil de Comunicação
1	Mensagens rápidas	GOOSE/GSSE
1A	Trip	GOOSE/GSSE
2	Mensagens de média velocidade	Cliente/Servidor
3	Mensagens lentas	Cliente/Servidor
4	Rajada de dados (<i>raw data messages</i>)	SV
5	Funções de transferência de arquivo	Cliente/Servidor
6	Mensagens de sincronismo de tempo	<i>Time Sync</i>

Fonte: Adaptado de IEC-61850-8-1 (2004).

As mensagens do tipo 1 e 1A possuem restrições críticas de tempo, como por exemplo, o comando de abertura de um disjuntor de alta tensão, assim, necessitando maior prioridade para acessar a camada de enlace do modelo OSI. Já as mensagens do tipo 2, 3 e 5 permitem utilizar toda a pilha de protocolos, operando, por exemplo, sobre o TCP/IP. O tipo 4 é utilizado para transmissão de valores amostrados, os quais representam uma grande quantidade de dados, e o tipo 6 é usado para sincronismo de tempo entre os diferentes IEDs (MIRANDA, 2009).

No modelo OSI apenas as duas primeiras camadas, física e de enlace, são comuns a todos os serviços. As mensagens que possuem restrições quanto a atrasos são mapeadas diretamente para camada de enlace, enquanto que as demais, que não têm restrição de tempo, utilizam toda a pilha de protocolos. A Figura 9 apresenta uma visão geral dos padrões e protocolos utilizados pelos perfis de comunicação definidos na norma.

Para os serviços de mensagens GOOSE, GSSE e SV, que tem o objetivo de evitar latências no processo de transmissão, a comunicação é do tipo editor (*publisher*) e subscritor (*subscriber*). Neste tipo de comunicação não existe nenhum tempo gasto estabelecendo conexão ou de negociação para iniciar uma seção de comunicação. O emissor publica as mensagens e os usuários subscritos capturam a mensagem que tem interesse. Trata-se de mensagens *multicast* que podem ser recebidas e utilizadas por diferentes dispositivos na rede (PAULINO; SIQUEIRA; CARMO, 2010).

Os serviços cliente/servidor são especificados para operar sobre as camadas do modelo ISO e compatíveis com os perfis de comunicação do TCP/IP, sendo mapeados para o protocolo MMS definido na norma ISO-9506 (MIRANDA, 2009).

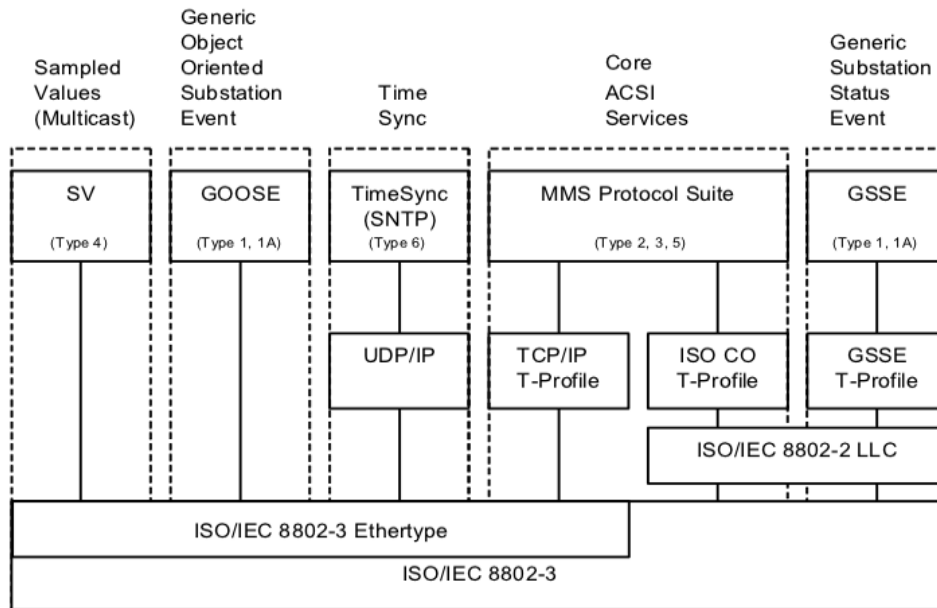


Figura 9: Protocolos e Perfis de Comunicação
 Fonte: IEC-61850-8-1 (2004).

Apesar dos IEDs possuírem relógio interno, para garantir a sincronização entre os IEDs, mensagens de sincronismo de tempo são enviadas utilizando o protocolo SNTP. A sincronização é essencial para qualquer sistema de análise que utilize informações coletadas nos IEDs.

3.8 TESTES DE CONFORMIDADE

A parte 10 da norma especifica as técnicas padrões para testes de conformidade de implementação, assim como técnicas de medição específicas para serem aplicadas quando declarando parâmetros de desempenho (IEC-61850-10, 2005). Entretanto, os testes de conformidade definidos não substituem os testes de aceitação específicos relacionados a aplicação desejada, como o FAT (*Factory Acceptance Test*) e o SAT (*Site Acceptance Test*), os quais devem representar todo o ambiente de operação e parametrizações específicas a serem utilizadas.

O objetivo da parte 10 da norma é assegurar que todos os seus modelos e serviços sejam executados corretamente, buscando possibilitar a interoperabilidade entre os dispositivos do sistema de automação. Nesse contexto, um sistema de testes baseado na IEC-61850 deve permitir um ensaio apropriado, adequado às exigências do sistema de supervisão, controle e proteção, e simular as características da subestação e do sistema elétrico em questão (MIRANDA, 2009).

4 RESULTADOS E DISCUSSÃO

Conforme exposto no capítulo anterior, a norma IEC-61850 aborda e especifica diversos aspectos fundamentais para a construção de aplicações voltadas para o segmento de automação de subestação. Alguns desses aspectos influenciam diretamente projetos de modelagem de *software*, como por exemplo, o Modelo de Dados e a Interface Abstrata de Serviços de Comunicação. Entretanto, apesar da amplitude da norma ser bastante abrangente, a mesma não foca em detalhes de implementação inerentes a construção de aplicações compatíveis com a normativa.

Durante um projeto de desenvolvimento de *software*, detalhes que talvez não são relevantes sob um ponto de vista de análise do problema, tornam-se fundamentais para a construção de uma aplicação que visa solucionar ou automatizar o problema em análise. Entre estes detalhes, pode-se destacar a definição da arquitetura base do *software*. Esta arquitetura representa como será organizado a estrutura lógica do *software*, dos processos, componentes e suas dependências e relacionamentos. A definição de uma arquitetura base também será um fator relevante para garantir que os modelos conceituais propostos na norma sejam corretamente implementados na aplicação em desenvolvimento, contribuindo assim para que o objetivo principal da IEC-61850 seja alcançado, ou seja, assegurar a interoperabilidade entre diferentes aplicações e equipamentos.

Os requisitos ou funcionalidades que tiveram seu comportamento descrito através de modelos conceituais podem ser descritos ou modelados em termos de classes, métodos, estruturas de dados, etc, chamados de componentes de *software*. Estes componentes, interagindo entre si e com outros componentes, permitem implementar os requisitos do modelo conceitual. A organização e estrutura desses componentes e seus inter-relacionamentos representa a arquitetura base do *software*, servindo como guia para o desenvolvimento, manutenção e evolução das aplicações ao longo do tempo.

4.1 ASPECTOS NÃO ESPECIFICADOS NA IEC-61850

Com base nos estudos e análises realizados sobre a norma IEC-61850, correspondente ao escopo deste trabalho, foram identificados alguns aspectos que não estão definidos no modelo conceitual proposto pela normativa. Estes aspectos estão listados a seguir:

- Estrutura de controle para receber, identificar e invocar os serviços de mensagens cliente/servidor (ACSI);
- Estrutura de controle para receber, identificar e processar mensagens com restrições tem-

porais (GOOSE);

- Estrutura para detectar, disparar e processar eventos;
- Estrutura para atualizar as classes de dados dos nós lógicos com os valores aferidos em sensores de equipamentos primários no nível de processo;
- Estrutura para aplicar nas saídas digitais dos dispositivos físicos os valores correspondentes atualizados por meio dos atributos de dados dos nós lógicos;
- Estratégia de alocação de *threads* e processos de sistema operacional;
- Estrutura de pacotes para modularização e organização de componentes e código fonte.

Nas seções seguintes apresenta-se como os aspectos citados acima, unidos ao modelo conceitual da norma, podem ser modelados em termos de componentes de *software* e como estruturar estes componentes de modo a definir a arquitetura base do *software*.

4.2 ESTRUTURA GERAL DA ARQUITETURA PROPOSTA

Para caracterizar o ambiente de *hardware* e *software*, bem como a comunicação de dados existente em uma subestação elétrica, no contexto de automação de subestação, considerando os requisitos gerais de comunicação definidos na norma IEC-61850, apresenta-se na Figura 10 um diagrama ilustrativo da estrutura básica de comunicação entre IEDs e sistemas em uma subestação.

Através do diagrama da Figura 10 nota-se que a comunicação entre IEDs ocorre essencialmente por meio de mensagens GOOSE, ou seja, mensagens com restrições temporais e não orientadas a conexão, devido a necessidade de respeitar limites de tempo para processamento e transmissão. Essas mensagens trafegam sobre a rede do barramento de processo. Já a conexão com os equipamentos primários da subestação, como por exemplo, disjuntores e transformadores de tensão e corrente, normalmente ocorre por meio de cabos físicos (*hardwired*) ligados ao equipamento primário e às portas correspondentes disponibilizadas por placas de aquisição de dados conectadas ao IED. Por outro lado, a comunicação com o sistema supervisor, o qual normalmente está localizado na sala de controle e disponibiliza recursos de interface com o usuário, ocorre por meio de mensagens do protocolo MMS. Esta comunicação é orientada a conexão e não apresenta restrições temporárias críticas, utilizando como meio físico a rede do barramento de estação.

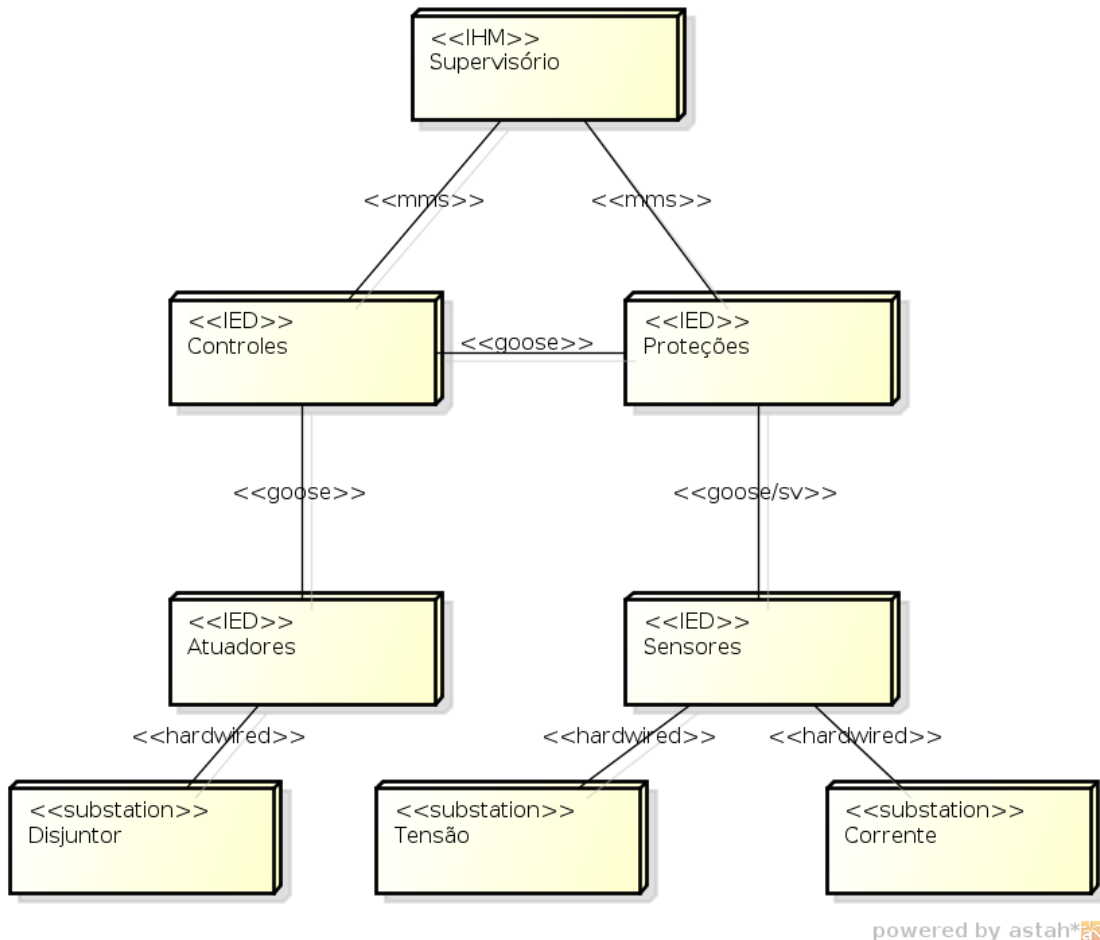


Figura 10: Estrutura de Comunicação em uma Subestação Elétrica

Internamente cada IED deve respeitar um modelo conceitual de dados e atributos especificados na norma. Esse modelo conceitual é fundamental para possibilitar a interoperabilidade entre diferentes IEDs. Uma representação simplificada do modelo conceitual é apresentado no diagrama UML da Figura 11.

No diagrama da Figura 11 as classes estão representadas com nomes escritos em maiúsculo, significando que trata-se de classes previstas e definidas na própria normativa. As classes a direita entre SERVER e DATA-ATTRIBUTE representam essencialmente o modelo abstrato dos dados e atributos contidos e manipulados por um IED. Atualmente a norma define 92 nós lógicos e 355 classes de dados, quantidades que tendem a aumentar ao longo do tempo. Esta característica de padronização tem a finalidade de facilitar a interoperabilidade, permitindo conhecer a capacidade de cada IED de acordo com os nós lógicos e classes de dados alocados e configurados para o mesmo. As classes de blocos de controle (CONTROL-BLOCK) são estruturas utilizadas essencialmente para log e envio de subconjuntos de dados (DATA-SET) para outros IEDs, sendo esta ação normalmente disparada por eventos internos ou periodicamente, dependendo da classe de controle e das configurações definidas para o equipamento. Do ponto de

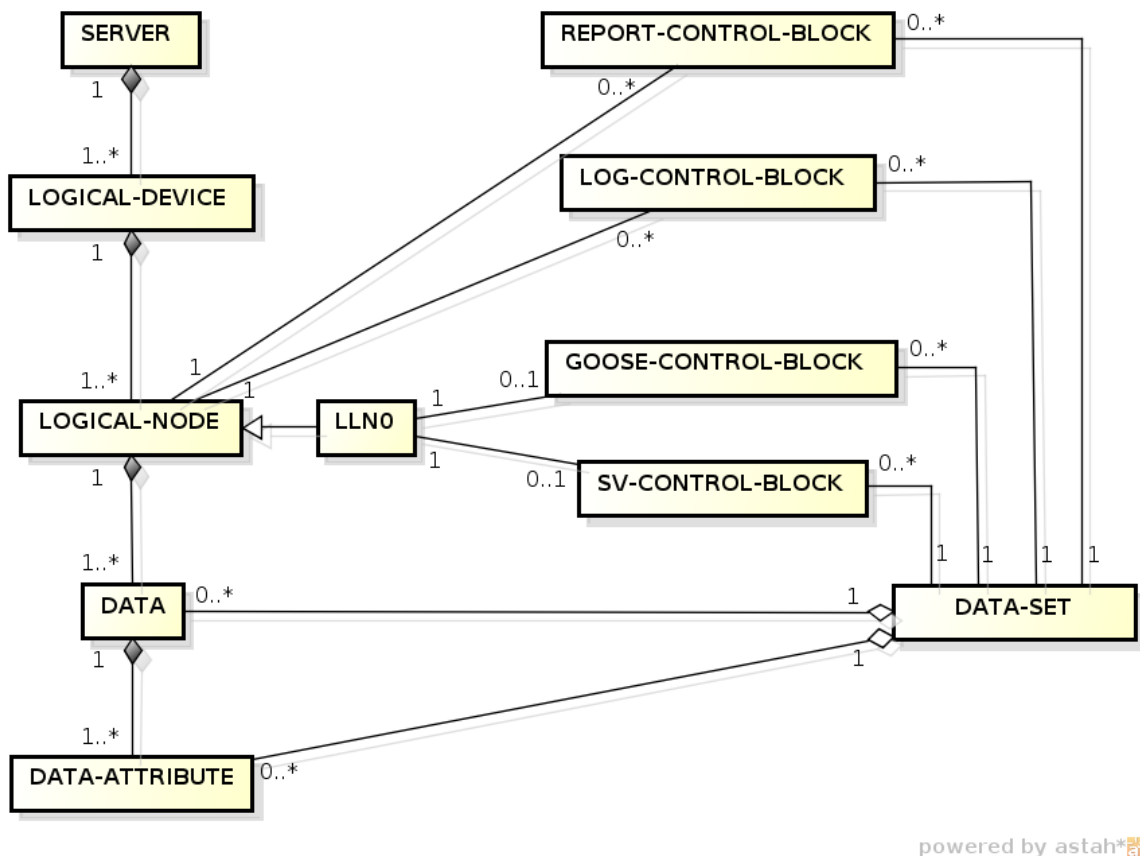


Figura 11: Modelo Conceitual da Norma IEC-61850

vista de um IED receptor, a chegada de um subconjunto de dados pode disparar eventos internos e/ou atualizar dados de nós lógicos do IED receptor, dependendo de configurações de mapeamento específicas para esta finalidade.

Para suportar a implementação real do modelo conceitual, estruturas internas específicas são necessárias. Essas estruturas podem ser organizadas e representadas em componentes de *software*, onde cada componente possui características e funções específicas. No diagrama da Figura 12 apresenta-se a estrutura de componentes proposta neste trabalho. O componente **SERVER** encapsula os modelos de dados e atributos do respectivo IED, respeitando o modelo conceitual previsto na normal. Os componentes **Control Blocks** e **DATA-SET** encapsulam a implementação das classes de controle e dos subconjuntos de dados. Todos estes componentes implementam e disponibilizam serviços específicos da interface **ACSI**, conforme definido na norma IEC-61850. Além dos serviços previstos na interface **ACSI**, outros serviços adicionais são necessários para tratamento de eventos e controles internos, gerando uma dependência entre os três grupos de componentes conforme representado no diagrama. Posteriormente, através de diagramas de sequência, será ilustrado como estes controles internos poderão ser implementados.

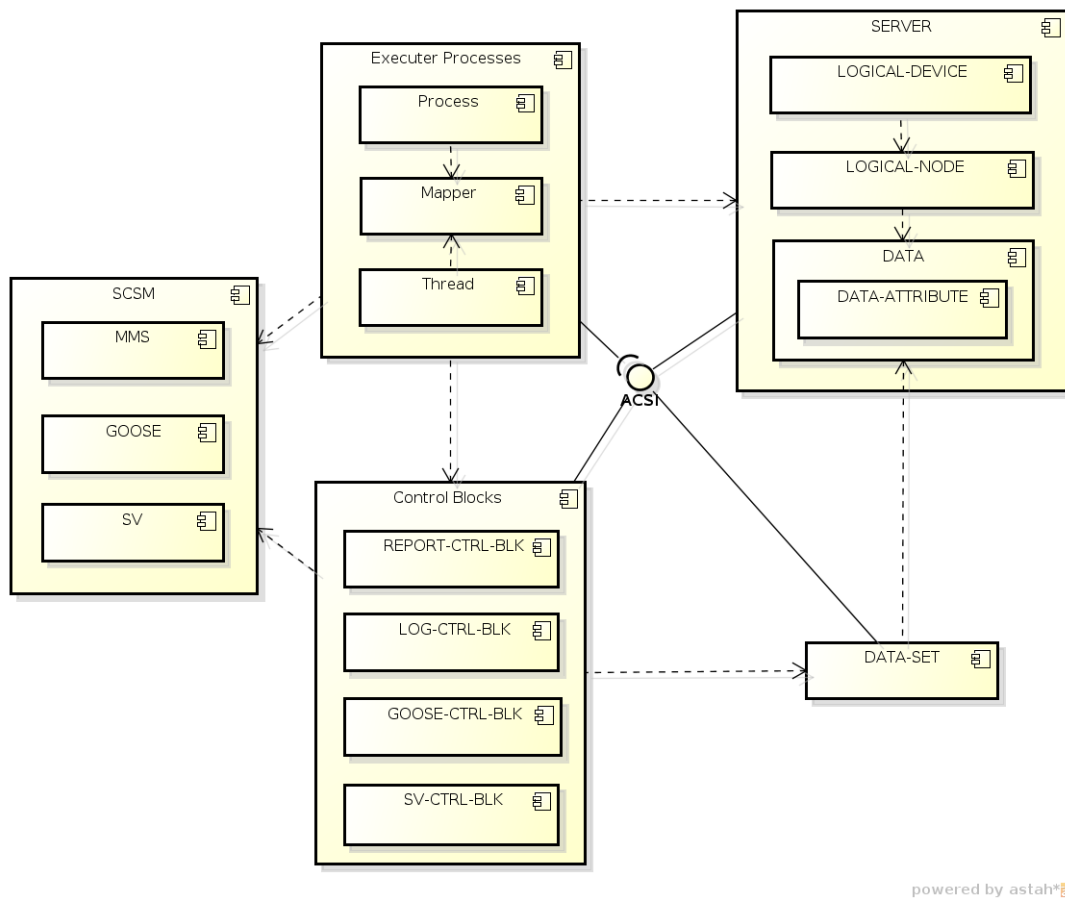


Figura 12: Estrutura Lógica de Componentes de um IED

Ainda no diagrama da Figura 12 está representado o componente SCSM responsável por implementar os perfis e protocolos específicos da rede de comunicação, abstraindo para os demais componentes o meio físico e detalhes de implementação da rede de comunicação. Os componentes do Control Blocks utilizam o componente SCSM para implementar serviços da interface ACSI que enviam informações para a rede de comunicação. O componente Executer Processes encapsula os processos de sistema operacional e *threads* de aplicação necessários para realizar todo o processamento interno do IED. Alguns desses processos e *threads* executam serviços com restrições temporais, conforme previsto pela norma, exigindo assim que o sistema operacional utilizado tenha suporte à construção de aplicações de tempo real. No modelo de componentes apresentado, os “processos” representam os casos onde haverá uma única instância daquele componente de processamento para todo o IED, enquanto que as “*threads*” significam que poderá haver várias instâncias do mesmo componente de processamento, como ocorre, por exemplo, no caso do processamento de diferentes conexões MMS estabelecidas com o IED.

Os processos/*threads* monitoram e aguardam a ocorrência de eventos que disparam a execução de serviços específicos e/ou atualização de atributos de dados dos nós lógicos. A referência

para o serviço e/ou atributo de dado correspondente é obtida através de um componente de mapeamento que possui acesso às instâncias de classes e seus respectivos serviços. Os eventos podem ser, por exemplo, a chegada de mensagens via rede de comunicação, a mudança de estado em canais de aquisição de dados, o disparo periódico de um temporizador, ou até mesmo eventos derivados de outros eventos, como ocorre no caso de atualizações de subconjuntos de dados (DATA-SET).

A norma IEC-61850 prevê mais de noventa nós lógicos, entretanto, conceitualmente, um nó lógico pode ser alocado à qualquer IED e pode ser configurado para interagir com qualquer outro nó lógico, de qualquer IED. O modelo de componentes proposto leva em consideração esses aspectos, de modo que a mesma estrutura pode ser utilizada para implementar IEDs com diferentes nós lógicos e com diferentes configurações de interação.

Considerando os requisitos gerais da norma e as características e estruturas discutidas, apresenta-se no diagrama da Figura 13 a estrutura geral de pacotes sugerida para modularização e organização do código fonte e componentes, para o caso de uma futura implementação do modelo proposto.

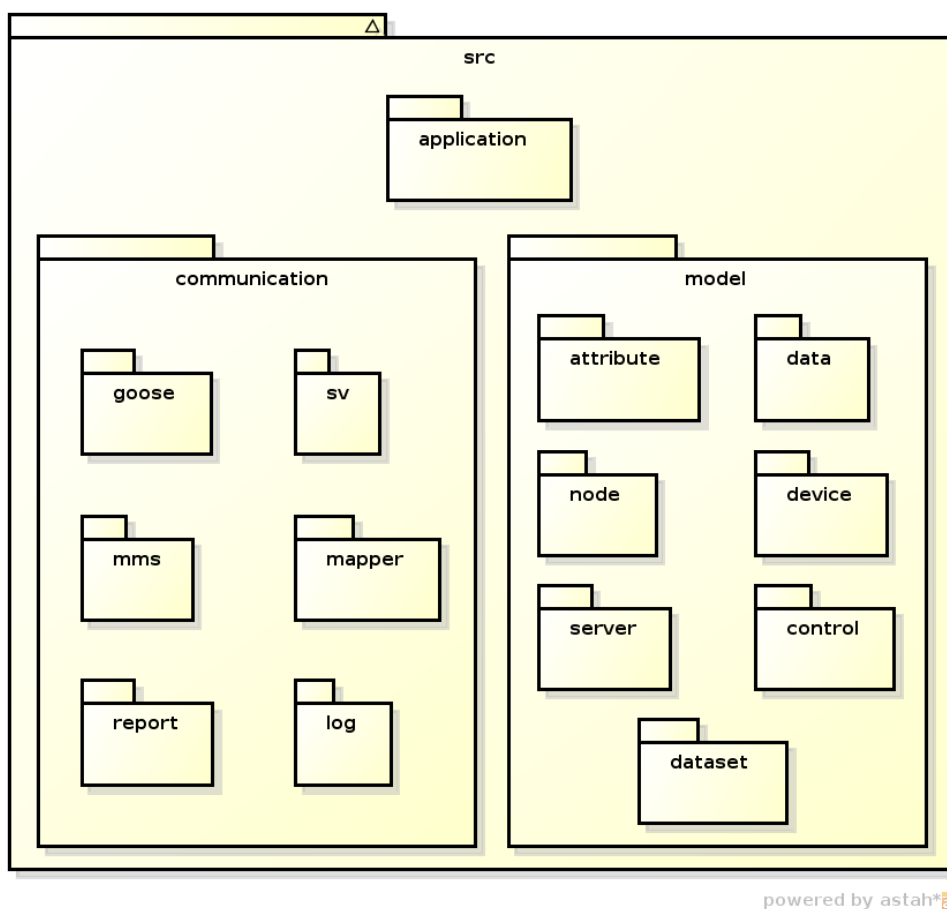


Figura 13: Estrutura de Pacotes para Código Fonte e Componentes

Conforme ilustrado na Figura 13, sugere-se uma divisão lógica de pacotes entre os processos/*threads*, componentes e classes relacionados com o processamento e comunicação de dados, pacote *communication*, e os blocos de controle, componentes e classes relacionados com o modelo de dados, pacote *model*. Essa divisão reduz o impacto sobre a implementação do modelo de dados frente à possíveis alterações e atualizações evolutivas na camada e componentes de comunicação. Outro pacote, *application*, é sugerido para organizar classes e componentes que não estejam relacionados com o modelo de dados ou de comunicação, como por exemplo, classes e componentes internos, não especificados na norma, relacionados à aquisição de dados, a leitura e processamento de arquivos da linguagem SCL e a inicialização do sistema.

Nas seções seguintes serão apresentadas com mais detalhes as classes e estruturas sugeridas para a arquitetura base do sistema, facilitando assim o entendimento do relacionamento entre os pacotes sugeridos e as respectivas classes e componentes propostos neste trabalho.

4.3 ESTRUTURA DE PROCESSOS E *THREADS* PARA A ARQUITETURA PROPOSTA

Internamente um IED, dependendo de suas configurações e alocação de nós lógicos, deverá executar diversas tarefas paralelamente, entre elas:

- Leitura de dados amostrados e atualização de estruturas lógicas internas;
- Detecção e processamento de eventos disparados;
- Processamento de mensagens não orientadas a conexão recebidas via rede de comunicação;
- Envio de mensagens não orientadas a conexão para a rede de comunicação;
- Monitoramento de solicitações de abertura de novas conexões MMS;
- Tratamento e processamento de conexões MMS abertas;
- Envio periódico de dados amostrados para a rede de comunicação;
- Envio contínuo de “relatórios” de alteração de dados (*reports*) para a rede de comunicação;
- Registro contínuo de logs de alteração de dados.

Para atender a demanda de processamento de um IED, apresenta-se no diagrama da Figura 14 a estrutura de processos e *threads* proposta. Os estereótipos de “processo” (*process*) representam casos onde haverá uma única instância daquele componente de processamento, enquanto que os estereótipos “thread” significam que poderá haver várias instâncias, dependendo de configurações do IED. Os estereótipos cujo nome contém os caracteres “RT” significam que executam processamento com restrições temporais, necessitando, para uma futura implementação, de recursos de sistemas de tempo real.

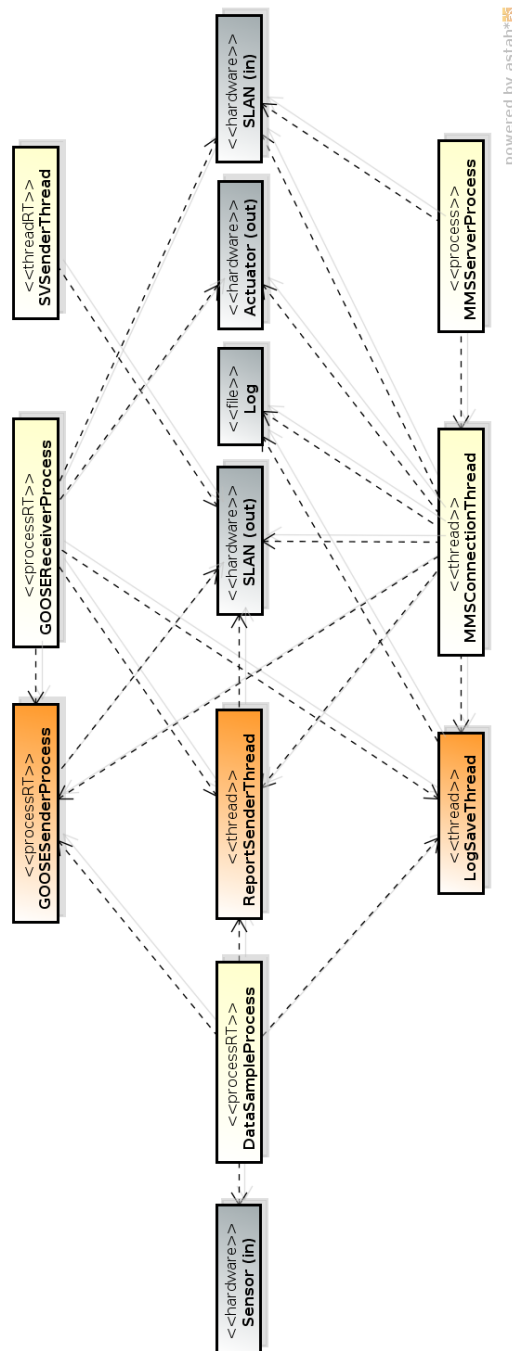


Figura 14: Dependências dos Processos e *Threads* de um IED

No diagrama da Figura 14 estão representadas as dependências entre processos, *threads* e itens de *hardware* como sensores, atuadores e conectores de rede. Posteriormente, por meio de diagramas de sequência, será ilustrado como ocorre a interação entre os vários processos e *threads*. Na lista a seguir estão descritas as principais atribuições de cada processo/*thread*, sendo importante considerar que a execução de uma determinada tarefa por um processo ou *thread* pode causar o disparo de eventos que somente serão processados por outros processos ou *threads*, facilitando assim a mensuração do tempo de processamento de cada tarefa, característica importante para sistemas de tempo real.

- **MMServerProcess**: Responsável por receber pedidos de novas conexões MMS e criar uma *thread* específica para tratar cada nova conexão;
- **MMSConnectionThread**: Responsável por receber e processar as requisições solicitadas por meio da conexão MMS vinculada, como por exemplo, a execução de serviços da interface ACSI;
- **SVSenderThread**: Responsável por enviar valores amostrados para a rede de comunicação, respeitando a periodicidade definida para cada instância;
- **GOOSEReceiverProcess**: Responsável por receber e processar mensagens GOOSE, as quais podem causar a atualização de atributos de dado e/ou a ativação de canais de saída digital;
- **DataSampleProcess**: Responsável por ler os dados amostrados pela placa de aquisição e atualizar os atributos de dados correspondentes;
- **GOOSESenderProcess**: Responsável por processar eventos disparados com a finalidade de enviar mensagens GOOSE para a rede de comunicação;
- **ReportSenderThread**: Responsável por processar eventos disparados com a finalidade de enviar “relatórios” de alteração de dados (*reports*) para a rede de comunicação usando conexões previamente estabelecidas, respeitando os parâmetros configurado para cada instância;
- **LogSaveThread**: Responsável por processar eventos disparados com a finalidade de registrar logs de alteração de dados, respeitando os parâmetros configurado para cada instância. Posteriormente os logs poderão ser acessados através de serviços da interface ACSI.

Para facilitar a compreensão, uma representação simplificada do diagrama da Figura 14 é apresentada na Figura 15, ilustrando apenas as dependências entre os processos e *threads* que podem disparar eventos, normalmente gerados pela atualização de atributos de dado, e os processos e *threads* que processam os eventos disparados.

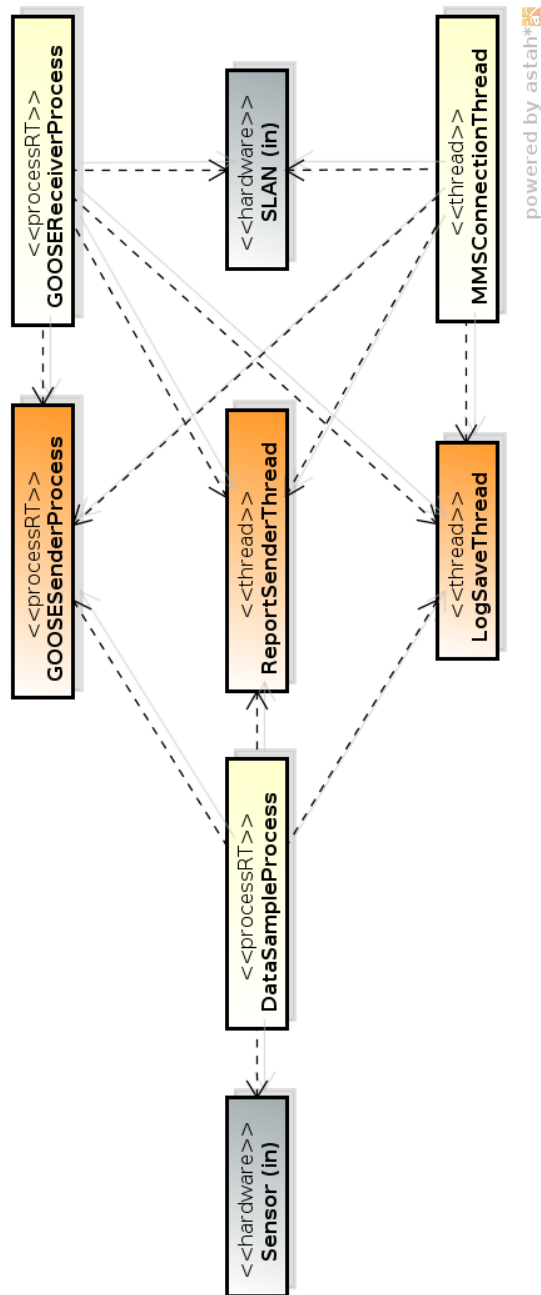
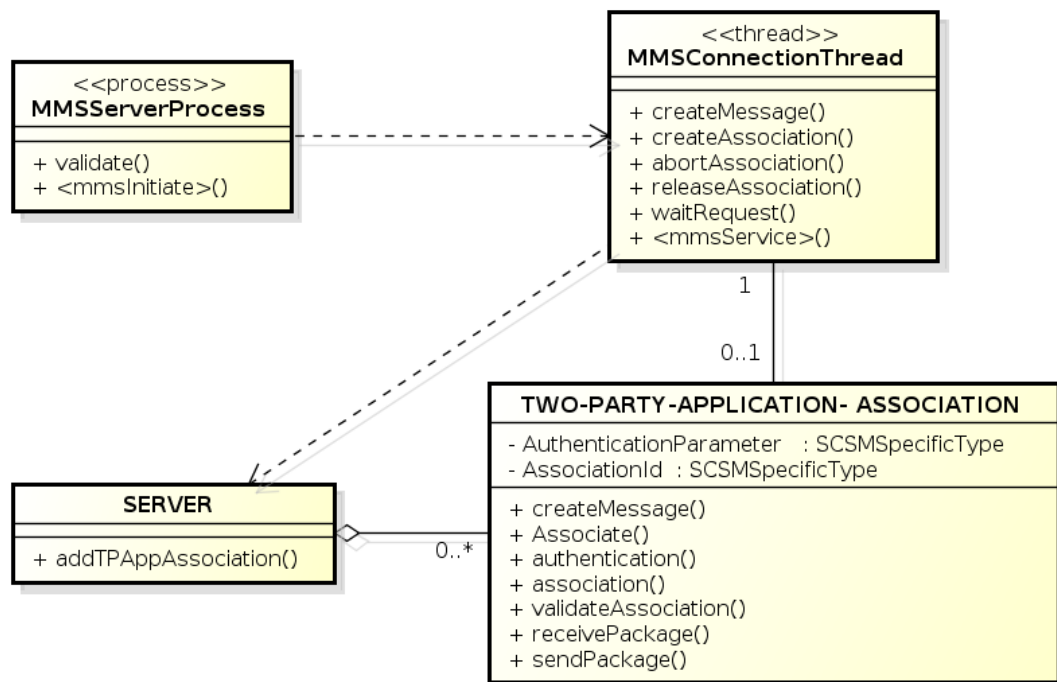


Figura 15: Dependências dos Processos e *Threads* para Disparo de Eventos

4.4 CLASSES ADICIONAIS E SEQUÊNCIAS DE EXECUÇÃO SUGERIDAS PARA A ARQUITETURA PROPOSTA

Os diagramas a seguir ilustram como poderá ser implementado os componentes, classes e controles internos do IED, ilustrando também as interações entre classes previstas no modelo conceitual e classes adicionais necessárias para implementar controles internos não especificados na normativa. As classes representadas com nomes escritos em maiúsculo são definidas pela própria norma, as demais são classes adicionais propostas neste trabalho. De maneira análoga, os métodos cujo nome inicia com letra maiúscula representam operações já previstas e especificadas pela norma.

Nas Figuras 16 e 17 está modelado a estrutura necessária para implementar o processo de abertura de uma nova conexão MMS entre um IED servidor e um sistema supervisorio. Nesse modelo, a classe `MMServerProcess` monitora a interface de rede e aguarda a chegada de uma nova requisição para abertura de conexão, ou associação, como é chamado na terminologia da norma IEC-61850. Para cada nova conexão requisitada é criado uma instância da classe `MMSConnectionThread` para gerar os parâmetros de associação e autenticação, conforme definidos pela norma, ficando essa nova instância responsável por tratar todas as novas mensagens vinculadas à conexão estabelecida. A nova associação estabelecida também é inserida na classe `SERVER`, respeitando assim requisitos da norma.



powered by astah*

Figura 16: Diagrama de Classe para Criar uma Nova Associação Orientada à Conexão

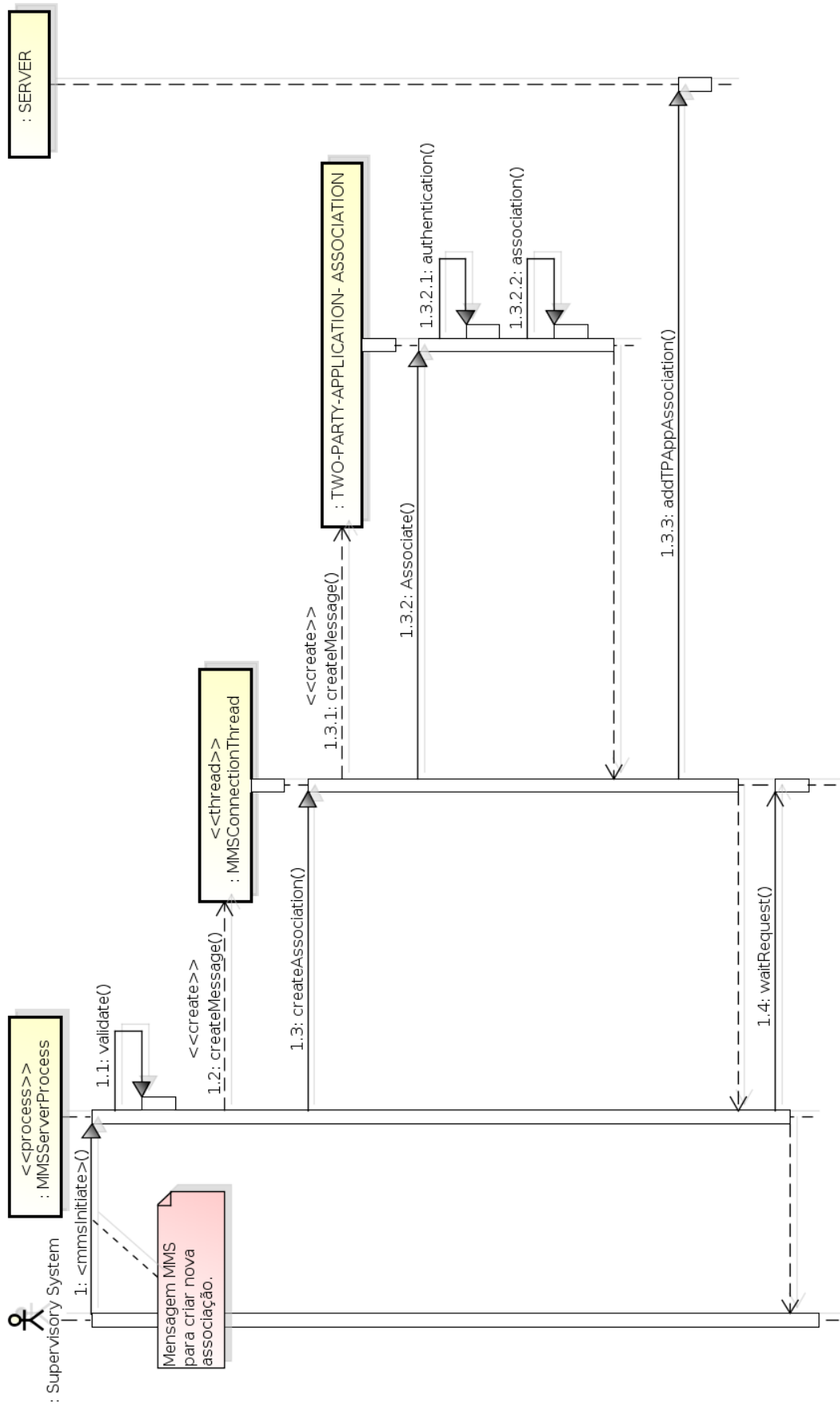
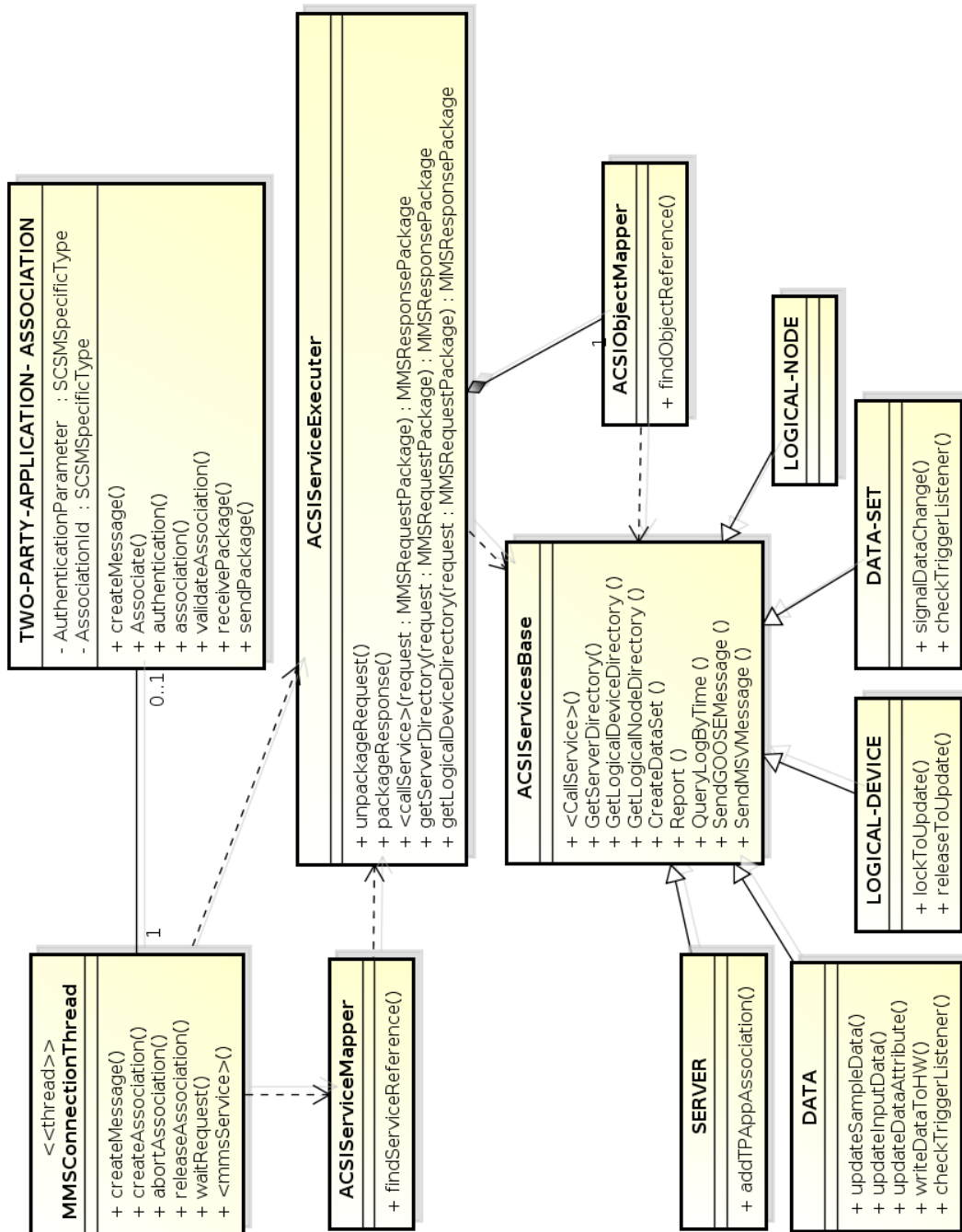


Figura 17: Diagrama de Sequência para Criar uma Nova Associação Orientada à Conexão

Com a conexão estabelecida a *thread* associada monitora a interface de rede e aguarda por requisições dos serviços da interface ACSI. Após a chegada de uma requisição de serviço, a qual é transmitida por meio de mensagens MMS, é necessário decodificar a mensagem recebida e identificar o serviço requisitado, bem como localizar a instância da classe (objeto) referenciada na mensagem MMS para então invocar o serviço correspondente daquele objeto. A norma IEC-61850 não especifica como este procedimento será realizado, assim, apresenta-se nos diagramas das Figuras 18 e 19 o modelo sugerido para implementação do procedimento citado.



powered by astah

Figura 18: Diagrama de Classe para Executar um Serviço da Interface ACSI

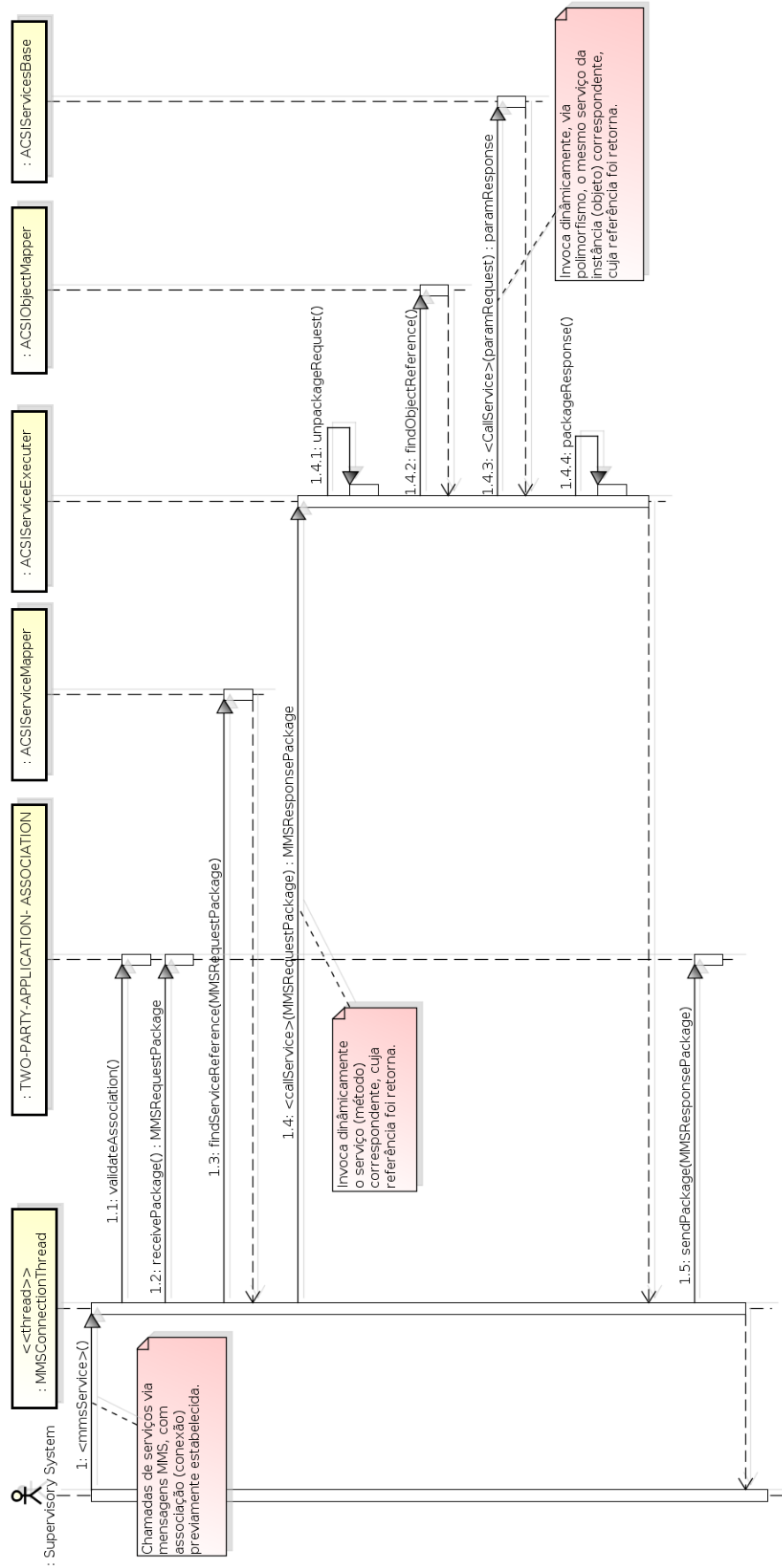


Figura 19: Diagrama de Sequência para Executar um Serviço da Interface ACSI

Conforme ilustrado nos diagramas das Figuras 18 e 19 foi criada uma estrutura de mapeamento que permite, dinamicamente, em tempo de execução, localizar a instância de objetos e serviços, de acordo com os parâmetros recebidos por meio da mensagem MMS. Nessa estrutura a *thread* de processamento `MMSConnectionThread` utiliza a classe `ACSIServiceMapper` para identificar qual é o serviço (método) invocado e obter a referência para esse serviço implementado na classe `ACSIServiceExecuter`. Os códigos específicos de cada serviço e não vinculados a instância do objeto referenciado na mensagem MMS, como por exemplo, códigos para empacotar e desempacotar parâmetros de entrada e de retorno, ficarão definidos somente na classe `ACSIServiceExecuter`.

Através da classe `ACSIObjectMapper` e do serviço invocado, a classe `ACSIServiceExecuter` localiza a instância do objeto referenciado na mensagem MMS e executa o método (serviço) correspondente. O método a ser executado no objeto referenciado é o mesmo que está sendo executado na classe `ACSIServiceExecuter`, permitindo assim que a chamada ao método seja resolvida em tempo de compilação. Todos os serviços da interface `ACSI` devem possuir um método abstrato definido na classe abstrata `ACSIServicesBase`, a qual deverá ser herdada por todas as classes que implementam serviços da interface `ACSI`, possibilitando assim o uso de polimorfismo na implementação e execução do serviço.

O processamento de requisições de serviços da interface `ACSI` é uma das tarefas executadas por um IED. Dependendo de configurações e alocação de nós lógicos, um IED deverá também realizar a aquisição de dados analógicos ou digitais amostrados via *hardware* específicos, como por exemplo, placas de aquisição de dados. O processamento de tarefas como essa ocorre em paralelo com outras tarefas executadas pelo IED. No modelo proposto, o processamento dos dados amostrados é realizado pelo processo `DataSampleProcess`, que tem a função de ler os dados amostrados em *hardware* específico e atualizar estruturas lógicas internas do IED, como por exemplo, as classes `DATA`. Durante a atualização das estruturas lógicas internas, alguns eventos podem ser disparados, conforme definido na norma IEC-61850. De acordo com o tipo de evento disparado será ativado, dinamicamente, um outro processo ou *thread* para realizar o processamento necessário para esse evento. Para manter a consistência dos dados vinculados ao evento, sem manter bloqueada a atualização das estruturas lógicas do equipamento antes do completo processamento do evento, é realizado uma cópia dos dados no instante em que o evento foi disparado. Essa cópia contém todos os dados necessários para posterior processamento do evento, sendo inserida em um *buffer* acessível pela classe que fará o processamento do evento.

Nos diagramas das Figuras 20 e 21 é ilustrado como foi projetado esse processo de leitura dos dados amostrados, atualização das estruturas lógicas internas e disparo de eventos para enviar mensagens GOOSE. Através dos diagramas nota-se que quando uma classe de dados in-

terna é atualizada, é verificado se há “ouvintes” registrados para os atributos atualizados, sendo então esses ouvintes informados sobre a atualização efetuada. Os ouvintes, pelo modelo conceitual da norma, são DATA-SET. Quando um DATA-SET for atualizado, será comunicado os ouvintes dos DATA-SET, que são, segundo o modelo conceitual, os blocos de controles, como por exemplo, o GOOSE-CONTROL-BLOCK. Os blocos de controle, através de classes adicionais não previstas no modelo conceitual da norma, executam o registro do evento e posterior processamento do mesmo. É interessante ressaltar que essa estrutura para disparo, registro e posterior processamento do evento será a mesma para quando ocorre uma atualização nas classes de dados interna, por outros meios, como por exemplo, através de serviços da interface ACSI.

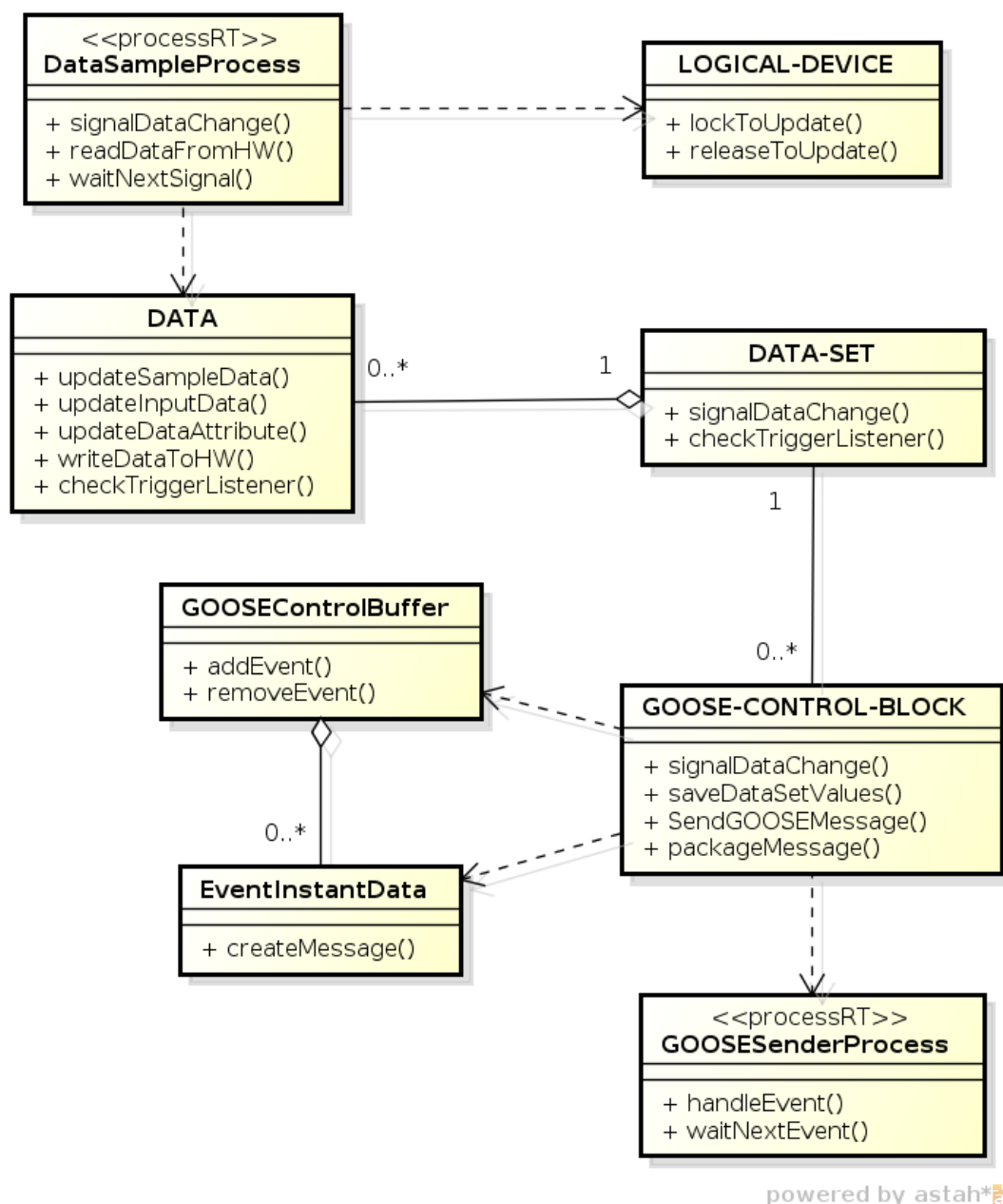


Figura 20: Diagrama de Classe para Disparar e Registrar um Evento GOOSE

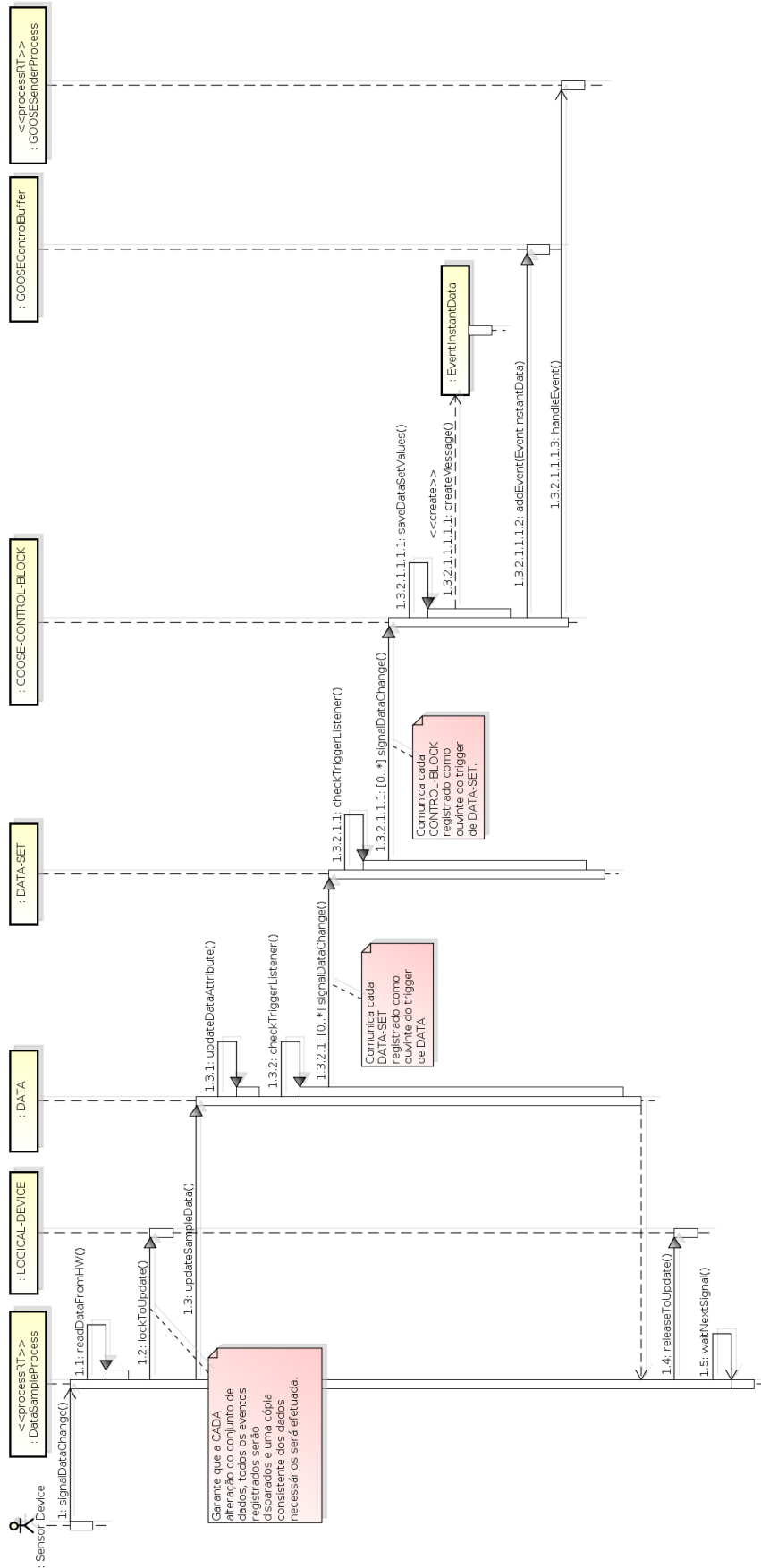
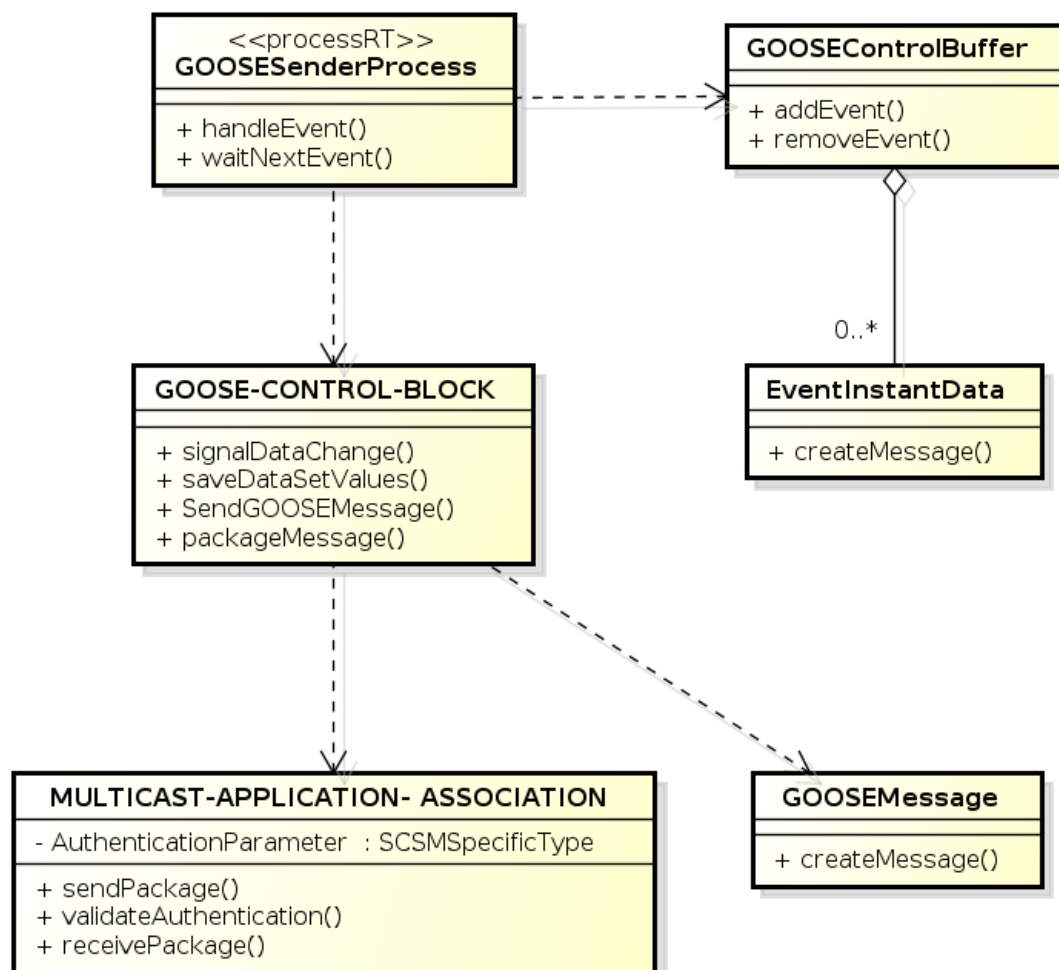


Figura 21: Diagrama de Sequência para Disparar e Registrar um Evento GOOSE

Após um evento ser disparado e registrado em um *buffer*, um outro processo ou *thread* será ativado para processar o evento. Nos diagramas das Figuras 22 e 23 é ilustrado o mecanismo de processamento de um evento para enviar uma mensagem GOOSE.

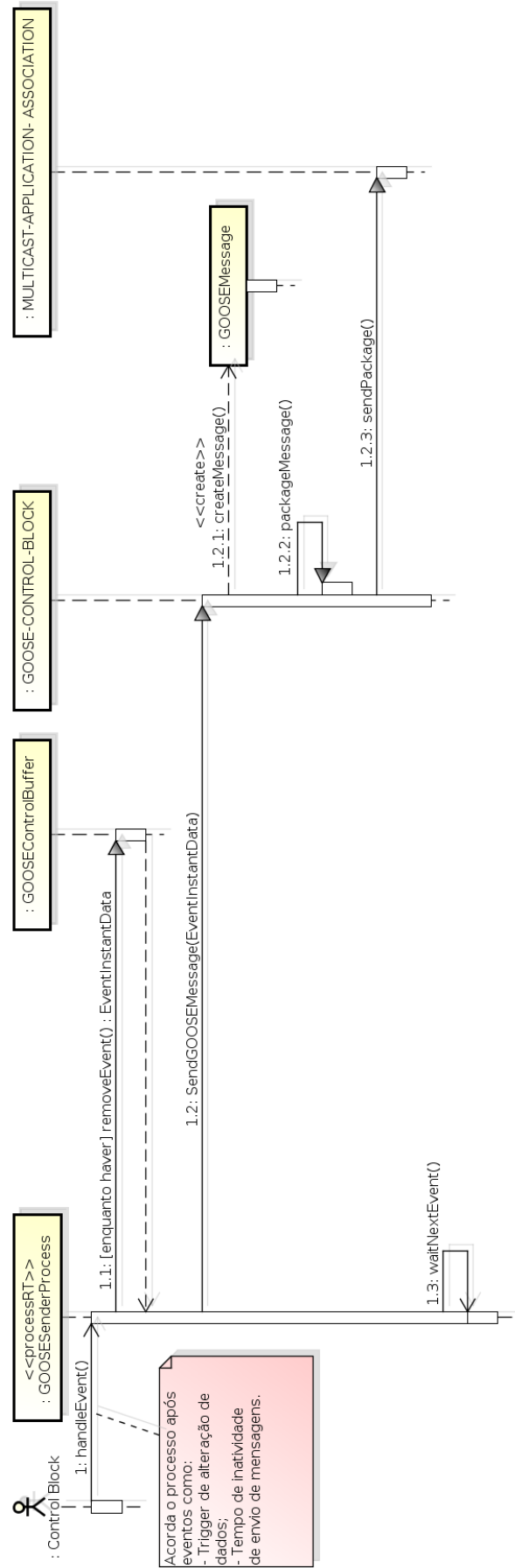
O processo `GOOSESenderProcess` é ativado quando haver um novo evento pendente para processamento inserido no *buffer*. Quando ativado o processo recupera os dados do evento armazenado na classe `GOOSEControlBuffer` e executa o serviço correspondente no bloco de controle. A classe do bloco de controle `GOOSE-CONTROL-BLOCK` instância e empacota uma nova mensagem GOOSE, de acordo com os critérios definidos na norma, e envia para a rede de comunicação utilizando a classe de associação `MULTICAST-APPLICATION-ASSOCIATION` vinculada ao bloco de controle.

Uma estrutura análoga a esta descrita acima também pode ser aplicada para o processamento de registros de logs e envio de *reports* de alteração de dados, os quais são executados pelas classes `LogSaveThread` e `ReportSenderThread`.



powered by astah*

Figura 22: Diagrama de Classe para Processar um Evento Enviando uma Mensagem GOOSE



powered by astah

Figura 23: Diagrama de Sequência para Processar um Evento Enviando uma Mensagem GOOSE

Dependendo das configurações definidas para a rede de comunicação entre os IEDs de uma subestação, em determinadas situações um IED pode ser o destinatário de eventos, recebendo, por exemplo, mensagens GOOSE geradas pelo processamento de um evento que ocorreu em um outro IED da rede. Nesse caso o IED destinatário deve possuir uma configuração de *input* para cada instância de nós lógicos que apresentarem essa característica, definindo nessa configuração um mapeamento entre os atributos de dados que podem ser recebidos via mensagem GOOSE e os atributos de dados das estruturas lógicas internas que serão atualizados com o valor recebido.

Entretanto, pelo modelo conceitual da norma, existem atributos de dados de determinados nós lógicos que representam, por exemplo, a saída digital do equipamento físico onde o dispositivo lógico (IED) se encontra. Assim, a atualização de um atributo de dado pode exigir que a saída digital de um determinado *hardware* seja simultaneamente atualizada. Além disso, conforme apresentado anteriormente, a atualização de um atributo de dado pode gerar o disparo de novos eventos internos no IED destinatário, caso existam “ouvintes” registrados para os atributos atualizados. A modelagem proposta para esse mecanismo está ilustrada nos diagramas das Figuras 24 e 25.

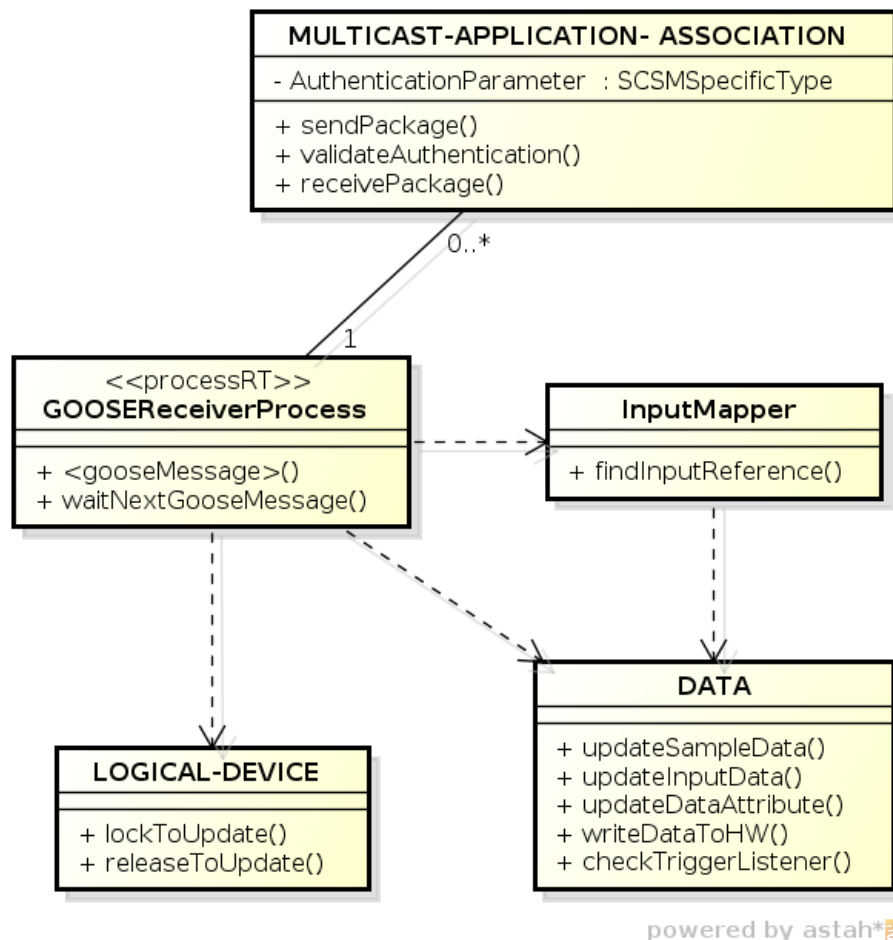


Figura 24: Diagrama de Classe para Receber e Processar uma Mensagem GOOSE

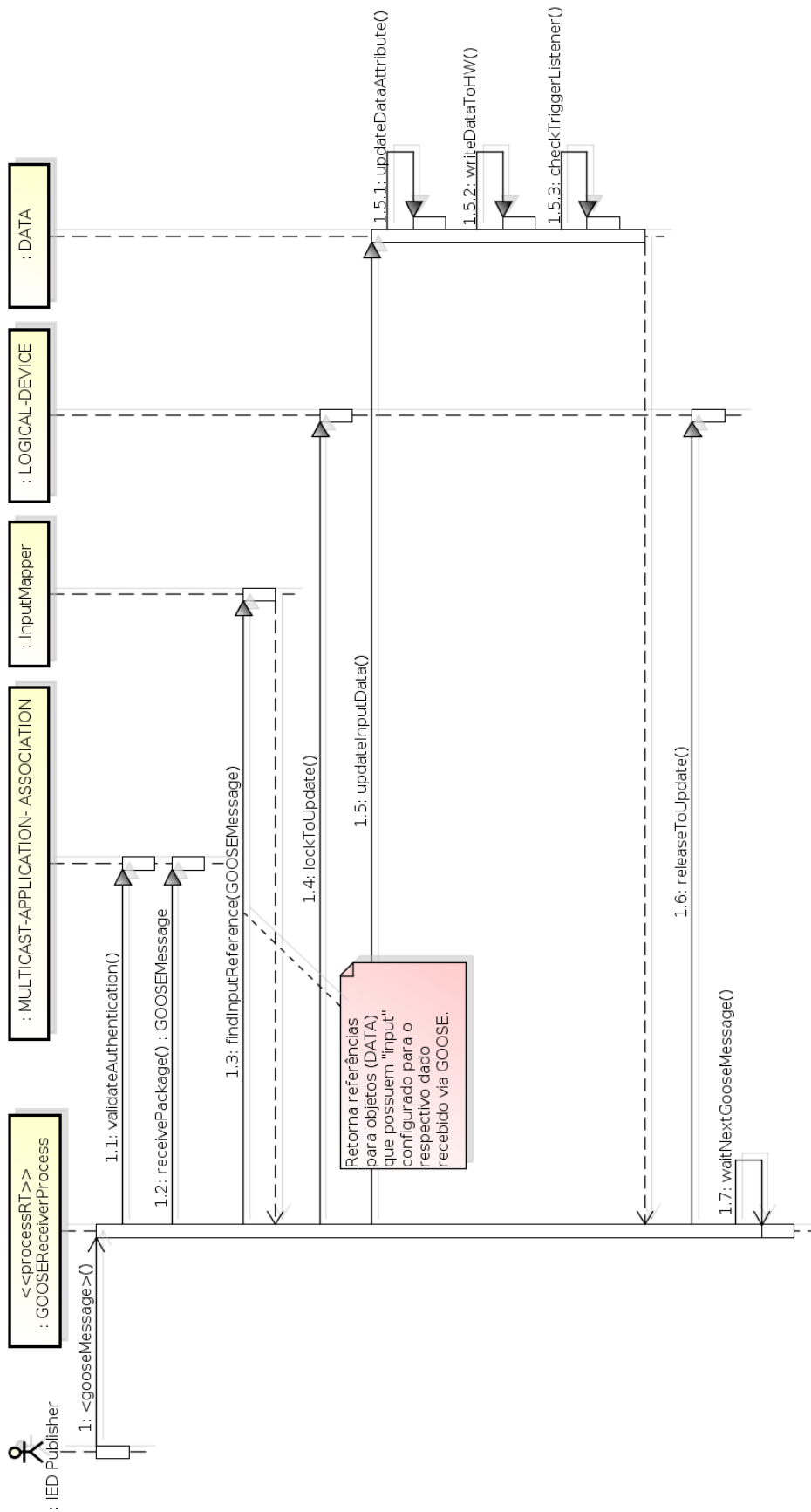


Figura 25: Diagrama de Sequência para Receber e Processar uma Mensagem GOOSE

5 CONSIDERAÇÕES FINAIS

5.1 CONCLUSÕES

A norma IEC-61850 representa um importante avanço no campo de comunicação e interoperabilidade entre IEDs de diferentes fabricantes, abrindo um novo horizonte para o desenvolvimento de soluções para sistemas de automação de subestações. A modelagem elaborada neste trabalho abordou, através de diagramas de sequência da UML, detalhes de implementação e implicações inerentes ao desenvolvimento de sistemas e que não foram explicitamente detalhados na normativa.

Comparando os modelos conceituais previstos na norma com os diagramas UML elaborados neste trabalho, pode-se observar que os diagramas apresentam um nível de detalhamento mais aprofundado, considerando operações e classes adicionais sugeridas neste trabalho como complemento às classes do modelo conceitual. As classes e operações adicionais sugeridas são necessárias quando abordamos o sistema de automação de subestação sob uma ótica de implementação de *software*.

A partir da modelagem apresentada, análises e discussões mais detalhadas sobre a viabilidade de implementação e confiabilidade da solução poderão ser realizadas, tanto no âmbito acadêmico como empresarial, possivelmente empregando abordagens e técnicas usadas na engenharia de *software* corporativos, permitindo assim a identificação de gargalos, pontos fortes e fracos, gerando como consequência o provável aprimoramento do modelo de arquitetura proposto.

Uma das principais dificuldades encontradas durante o desenvolvimento deste trabalho, em grande parte devido a abrangência e amplitude da norma, foi a identificação de quais detalhes já estavam especificados pela norma e quais não estavam, de modo a assegurar que os modelos propostos neste trabalho não conflitassem com os modelos conceituais especificados na normativa. Desta maneira, o presente trabalho contribuirá para que futuros projetos de implementação tenham um referencial para desenvolver o projeto detalhado do *software*, partindo de um modelo previamente estudado e definido, porém passível de aprimoramentos futuros, além de contribuir para a redução do tempo e esforço empregado à fase de engenharia de software.

Entretanto, para um melhor embasamento teórico sobre o assunto e as problemáticas, restrições e requisitos relacionados com a modelagem de sistemas para automação de subestações, é fundamental a leitura completa da série de documentos que compõem a norma IEC-61850 e correlatos, inclusive para um melhor entendimento deste trabalho, bem como da própria norma em si.

5.2 TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO

A modelagem proposta neste trabalho apresenta como alguns aspectos não especificados na IEC-61850 poderiam ser projetados e modelados de modo a definir a arquitetura base de um *software* para automação de subestação aderente à norma IEC-61850. Assim, para o desenvolvimento de trabalhos futuros, sugere-se o detalhamento dos diagramas, classes e componentes propostos, e a respectiva implementação do modelo detalhado visando a mensuração e comprovação da eficácia da solução proposta. Outro trabalho futuro possível seria a continuidade da análise da norma e identificação de outros aspectos que não estejam suficientemente detalhados na normativa, propondo, então, a modelagem para solucionar esses novos aspectos identificados, complementando e aprimorando o modelo de arquitetura base proposto neste trabalho.

REFERÊNCIAS

- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **The Unified Modeling Language User Guide**. 2. ed. NJ, USA: Addison-Wesley, 2005.
- GURJÃO, E. C.; CARMO, U. A.; SOUZA, B. A. Aspectos de comunicação da norma iec 61850. **Simpósio Brasileiro de Sistemas Elétricos**, Campina Grande, PB, jul. 2006.
- IEC-61850-1. **Communication Networks and Systems in Substations - Part 1: Introduction and Overview**. 1. ed. Geneva, Switzerland, 2003.
- IEC-61850-10. **Communication Networks and Systems in Substations - Part 10: Conformance Testing**. 1. ed. Geneva, Switzerland, 2005.
- IEC-61850-4. **Communication Networks and Systems in Substations - Part 4: System and Project Management**. 1. ed. Geneva, Switzerland, 2002.
- IEC-61850-5. **Communication Networks and Systems in Substations - Part 5: Communication Requirements for Functions and Device Models**. 1. ed. Geneva, Switzerland, 2003.
- IEC-61850-6. **Communication Networks and Systems in Substations - Part 6: Configuration Description Language for Communication in Electrical Substations related to IEDs**. 1. ed. Geneva, Switzerland, 2004.
- IEC-61850-7-1. **Communication Networks and Systems in Substations - Part 7-1: Basic Communication Structure for Substation and Feeder Equipment - Principles and models**. 1. ed. Geneva, Switzerland, 2003.
- IEC-61850-7-2. **Communication Networks and Systems in Substations - Part 7-2: Basic Communication Structure for Substation and Feeder Equipment - Abstract Communication Service Interface (ACSI)**. 1. ed. Geneva, Switzerland, 2003.
- IEC-61850-8-1. **Communication Networks and Systems in Substations - Part 8-1: Specific Communication Service Mapping (SCSM) - Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3**. 1. ed. Geneva, Switzerland, 2004.
- JOSÉ, N. M. **Definição e Implementação de Nós Lógicos de um Sistema de Automação de Subestação Baseado na Norma IEC 61850**. Monografia (Bacharelado em Ciência da Computação) — Universidade Estadual do Oeste do Paraná, Foz do Iguaçu, 2010.
- JUNIOR, P. S. P. et al. Iec 61850-9-2 avaliação e testes de um barramento de processos. **X STPC Seminário Técnico de Proteção e Controle**, Recife, out. 2010.
- KOPETZ, H. **Real-Time System. Design Principles for Distributed Embedded Applications**. Boston: Kluwer Academic Publishers, 1997.
- LACERDA, S. L. M.; CARNEIRO, G. H. R. Dispositivos eletrônicos inteligentes (ied's) e a norma iec 61850: União que está dando certo. **Congresso Norte-Nordeste de Pesquisa e Inovação**, João Pessoa, PB, nov. 2010.

- LOCKE, D. **What is Real-Time?** 2000. Disponível em: <<http://www.linuxdevices.com/articles/AT6090565653.html>>. Acesso em: 03 ago. 2011.
- MENDES, D. R. **Redes de Computadores - Teoria e Prática**. São Paulo: Novatec, 2007.
- MIRANDA, J. C. **IEC-61850: Interoperabilidade e Intercambialidade entre Equipamentos de Supervisão, Controle e Proteção Através das Redes de Comunicação de Dados**. Dissertação (Mestrado em Engenharia Elétrica) — Universidade de São Paulo, São Carlos, 2009.
- MODELO OSI. 2011. Disponível em: <http://pt.wikipedia.org/wiki/Modelo_OSI>. Acesso em: 10 maio 2011.
- OLIVEIRA, R. S. de; FARINES, J.; FRAGA, J. da S. **Sistemas de Tempo Real**. Florianópolis: UFSC, 2000.
- OMG. **Unified Modeling Language: Infrastructure**. 2.0. ed. Needham, MA, USA, 2005.
- OZANSOY, C. R.; ZAYEGH, A.; KALAM, A. The application-view model of the international standard iec 61850. **IEEE Transactions on Power Delivery**, v. 24, n. 3, p. 1132–1139, jul. 2009a.
- OZANSOY, C. R.; ZAYEGH, A.; KALAM, A. Object modeling of data and datasets in the international standard iec 61850. **IEEE Transactions on Power Delivery**, v. 24, n. 3, p. 1140–1147, jul. 2009b.
- PAULINO, M. E. C. Testes de ieds operando com redes de comunicação baseados na iec 61850. **Décimo Segundo Encontro Regional Ibero-americano do CIGRÉ**, Foz do Iguaçu, maio 2007.
- PAULINO, M. E. D. C.; SIQUEIRA, I. P. D.; CARMO, U. A. D. Requisitos para interoperabilidade de ieds e sistemas baseados na norma iec61850. **X STPC Seminário Técnico de Proteção e Controle**, Recife, out. 2010.
- PEREIRA, A. C. **Integração dos Sistemas de Proteção, Controle e Automação de Subestações e Usinas - Estado da Arte e Tendências**. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.
- PROUDFOOT, D. **UCA and 61850 for Dummies**. 2002. Disponível em: <[www.nettedautomation.com/download/UCA and 61850 for dummies V12.pdf](http://www.nettedautomation.com/download/UCA_and_61850_for_dummies_V12.pdf)>. Acesso em: 12 ago. 2011.
- SANTOS, L. F. D.; PEREIRA, M. Uma abordagem prática do iec61850 para automação, proteção e controle de subestações. **Simpósio de Automação de Sistemas Elétricos**, Salvador, ago. 2007.
- SCHWANKE, D. **Exame de Potenciais Evocados Auditivos utilizando Processador Digital de Sinais - DSPEA**. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.
- SMITH, S. W. **The Scientist and Engineer's Guide to Digital Signal Processing**. 1998. Disponível em: <<http://www.dspguide.com/whatdsp.htm>>. Acesso em: 02 ago. 2011.
- SOUTO, A. D. O.; FONSECA, M. D. O. Automação de subestações industriais. **Tecnologia em Metalurgia e Materiais**, São Paulo, v. 3, n. 3, p. 41–45, jan. 2007.

SPURGEON, C. E. **Ethernet: the definitive guide**. Sebastopol, CA, USA: O'Reilly, 2000.