

**Um Framework em Java de Tempo Real para a Construção de Aplicações
Baseadas nos Conceitos da Norma IEC 61850**

Projeto de Dissertação de Mestrado
Orientador: Rômulo Silva de Oliveira
Co-Orientadora: Patricia Plentz

1. Contexto

Os sistemas submetidos a requisitos temporais, também chamados de sistemas de tempo real, possuem requisitos que variam muito com relação ao tamanho, complexidade e criticalidade. Por exemplo, enquanto um mp3 player é relativamente simples, aplicações aviônicas e automotivas são extremamente complexas, possuindo dezenas de controladores eletrônicos interligados através de redes e barramentos de dados dedicadas.

No contexto da automação industrial, são muitas as possibilidades (ou necessidades) de empregar sistemas com requisitos de tempo real. Exemplos são os sistemas de controle embutidos em equipamentos industriais, os sistemas de supervisão e controle de células de manufatura e os sistemas responsáveis pela supervisão e controle de plantas industriais completas.

Na literatura os sistemas de tempo real são, em geral, classificados conforme a criticalidade dos seus requisitos temporais. Enquanto o não atendimento dos requisitos temporais nos sistemas de tempo real não críticos (soft real-time systems) resulta apenas na redução da utilidade destes sistemas (ou qualidade do serviço prestado), o não atendimento de um requisito temporal em sistemas de tempo real críticos (hard real-time systems) pode ter como resultado a perda de vidas humanas.

Nos sistemas em geral (que não são do tipo tempo real), a única preocupação é com a qualidade dos resultados. Embora uma execução rápida seja desejável, a abordagem é sempre do tipo "fazer o trabalho usando o tempo que for necessário". Sistemas tempo real possuem uma abordagem diferente, pois o tempo é limitado. É preciso garantir que será possível atender aos prazos, geralmente impostos pelo ambiente do sistema. Logo, a preocupação é "fazer o trabalho usando o tempo disponível". Esta preocupação tem como consequência um problema básico encontrado na construção de sistemas tempo real (críticos ou não): a alocação e o escalonamento das tarefas nos recursos computacionais disponíveis. Existe uma dificuldade intrínseca em compatibilizar dois objetivos fundamentais: garantir que os resultados serão produzidos no momento desejado e dotar o sistema de flexibilidade para adaptar-se a um ambiente dinâmico e, assim, aumentar sua utilidade.

Em um extremo (em geral associado a aplicações de tempo real críticas) existem soluções de escalonamento que supõe um conjunto fixo de tarefas a serem executadas. Estas soluções reservam recursos para o pior caso e são capazes de garantir que todas as tarefas serão concluídas no momento correto. Entretanto, aplicações construídas desta forma resultam em sistemas pouco flexíveis e na subutilização dos recursos computacionais. No outro extremo temos as soluções de escalonamento que não garantem o comportamento temporal da aplicação. Tarefas são escalonadas na medida do possível. Embora os recursos computacionais sejam plenamente utilizados e o sistema resultante seja bastante flexível, a falta de uma garantia prévia para o seu comportamento temporal

inviabiliza este tipo de solução para muitas classes de aplicações. A dificuldade de escalonar tarefas com requisitos de tempo real é bastante conhecida, constituindo uma área de pesquisa intensa atualmente.

1.1 IEC 61850

A função principal de um sistema elétrico de potência (SEP) é levar energia elétrica aos consumidores de forma segura, com qualidade e disponibilidade [1][2]. Os equipamentos principais incluem unidades geradoras, transformadores, disjuntores, seccionadoras, linhas de transmissão e linhas de distribuição. Um SEP poder ser dividido em três segmentos conhecidos como GTD: geração, transmissão e distribuição.

O termo automação é normalmente confundido quanto ao seu escopo real, talvez pela função mais geral de automatizar uma planta industrial como um todo, sendo muitas vezes usado para representar todo e qualquer sistema, incluindo sistemas mais específicos como proteção, controle ou monitoração. Para automação de subestações, por exemplo, o termo SAS (Substation Automation System) que inclui proteção, controle, monitoração, medicação e instrumentação, é comumente encontrado, o que corrobora com esta visão.

A especificação das interfaces entre esses subsistemas permitiria conhecer explicitamente suas interações e troca de dados. A norma IEC 61850 busca justamente criar um modelo comum de dados e objetos e definir interfaces nesse nível. A Figura 1 ilustra alguns elementos da norma, com destaque para os conceitos de “logical node” e “logical device”, que são centrais para o desenvolvimento de software segundo esta norma e fortemente orientados a objetos.

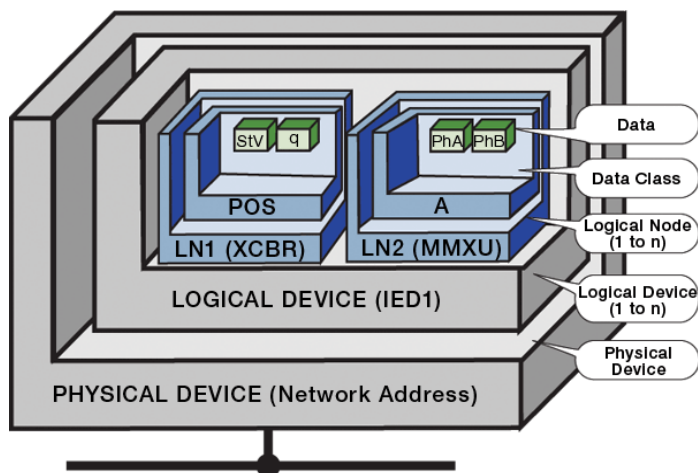


Figura 1 – Mapeando Nós Lógicos, Classes e Dados para o Mundo Real [3]

1.2 Real-Time Specification for Java

A linguagem Java tem provado ser um veículo apropriado para uma diversa gama de aplicações. Entre as várias razões dessa popularidade, destaca-se o suporte para concorrência através de mecanismos para sincronização e multithreads, gerência automática de memória através do mecanismo de coleta de lixo e a existência de uma máquina virtual Java (Java Virtual Machine - JVM) que permite aplicações Java executar sobre qualquer plataforma que suporte a JVM. Com isso, apesar das várias qualidades, o Java é criticado pela comunidade tempo real pela falta de previsibilidade de execução.

O Java Experts Group teve como objetivo definir uma especificação tempo real para Java (Real-Time Specification for Java – RTSJ), propondo estender a plataforma Java padrão, sem alterar o núcleo nem a sintaxe da linguagem. Esse grupo liberou a versão final da RTSJ em 2000. Para ser

considerada oficial, toda especificação Java necessita ser acompanhada de uma implementação de suas interfaces para servir como referência. A primeira implementação de referência foi submetida ao grupo de trabalho em meados de 2001 [4].

A RTSJ estende a plataforma Java através da criação de novas classes e interfaces e também através de mecanismos inseridos na máquina virtual Java (Java Virtual Machine – JVM). Mais especificamente, a RTSJ acrescenta à especificação Java existente os seguintes itens: [5]

- Threads Tempo Real. Estas threads possuem atributos de escalonamento que permitem definir tarefas tempo real (período de execução, deadline, tratadores de perda de deadline, prioridade, etc).
- Ferramentas e mecanismos que ajudam programadores escrever código tempo real Java sem a utilização do coletor de lixo, já que sistemas desse tipo buscam previsibilidade.
- Uma classe que trata eventos assíncronos e um mecanismo que associa eventos assíncronos com acontecimentos fora da JVM. Cada evento é atendido por uma thread, criada especialmente para essa função.
- Um mecanismo chamado transferência assíncrona de controle, que permite uma thread tempo real mudar o fluxo de controle em outra thread tempo real. Tal mecanismo é uma forma bastante controlada de uma thread tempo real lançar uma exceção em outra.
- Mecanismos que permitem os programadores acessar memória em endereços particulares.
- Mecanismos que permitem o programador controlar onde os objetos serão alocados na memória. A RTSJ criou dois novos domínios de alocação de memória para threads tempo real acessarem objetos: memória permanente e memória em escopos [6].

2. Objetivo

Desenvolver um framework em Java de Tempo Real para facilitar a implementação de aplicações que seguem os principais conceitos da norma IEC 61850. Esta norma é imensa, e cobre vários aspectos da automação de subestações elétricas. Entretanto, no contexto desta dissertação de mestrado, estamos preocupados apenas com os aspectos de orientação a objetos associados com o desenvolvimento de módulos de software, isto é, “logical devices” e “logical nodes”.

A norma utiliza orientação a objetos para descrever os módulos de software, e a existência de um framework em Java pode facilitar este desenvolvimento. Ao mesmo tempo, o emprego de Java de tempo real assegura um bom comportamento temporal, dado que o contexto em questão exige respostas rápidas e previsíveis.

Espera-se executar as aplicações contruídas através do framework proposto em PC normais, dado que é um ambiente de investigação e não uma aplicação real para ser colocada em campo (no caso, em uma subestação de energia elétrica).

A dissertação deverá incluir um levantamento bibliográfico sobre Java para tempo real e sobre as partes da norma IEC 61850 que utilizam orientação a objetos. A idéia de framework de software orientado a objetos também deverá ser estudada.

Espera-se ao final que o mestrando conheça a linguagem Java para tempo real, os princípios gerais da norma IEC 61850 e os detalhes relacionados com os conceitos de “logical node” e “logical device”, os quais são orientados a objetos. O estudo deverá considerar o desenvolvimento de um framework em Java que facilite o desenvolvimento de software conforme os conceitos citados antes.

3. Atividades

- (a) Estudar os conceitos básicos dos sistemas de tempo real.
- (b) Estudar a linguagem de programação Java e sua variante para tempo real.
- (c) Estudar o conceito de framework de software orientado a objetos.
- (d) Estudar a norma IEC 61850 nos seus aspectos de “logical node” e “logical device” e o desenvolvimento de software orientado a objetos.
- (e) Propor um framework em Java de tempo real que facilite o desenvolvimento de software conforme a norma IEC 61850.
- (f) Avaliar e melhorar o framework proposto a partir da experiência de implementação de algumas aplicações para teste.
- (g) Estabelecer os limites da tecnologia empregada no que se refere a execução de aplicações de tempo real, delimitando o escopo de requisitos temporais que podem ser atendidos com esta tecnologia.
- (h) Estabelecer um conjunto de recomendações e diretrizes para quem pretende usar o framework em questão.
- (i) Redigir artigos para publicação em congressos e/ou revistas.
- (j) Redigir a dissertação.

4. Bibliografia

- [1] Garcia, D. A.A.; Duzzi, F. E. Capítulo II - Tópicos de sistemas de transmissão e distribuição de energia elétrica. Disponível em: <http://www.iee.usp.br/biblioteca/producao/2012/Artigos%20de%20Periodicos/garciatopicos.pdf> e <http://www.osetoreletrico.com.br/web/component/content/article/80-distribuicao-de-energia/831-capitulo-ii-topicos-de-sistemas-de-transmissao-e-de-distribuicao-de-energia-eletrica.html>.
- [2] Garcia, D. A.A.; Duzzi, F. E. Capítulo I - Aspectos de sistemas de geração, transmissão e distribuição de energia. Disponível em: <http://www.iee.usp.br/biblioteca/producao/2012/Artigos%20de%20Periodicos/garciasaspectos.pdf> e <http://www.osetoreletrico.com.br/web/component/content/article/80-distribuicao-de-energia/796-capitulo-i-aspectos-de-sistemas-de-geracao-transmissao-e-distribuicao-de-energia.html>.
- [3] KEMA. A Primer on IEC 61850 grid data communications. Disponível em <http://www.dnvkemautilityfuture.com/a-primer-on-iec-61850-grid-data-communications>.
- [4] Plentz, P.D.M.; Montez, C.B.; Oliveira, R.O.; et al. Programação Baseada em Threads Distribuídas nas Especificações RT-CORBA 2.0 e Distributed RTSJ. In: WORKSHOP DE TEMPO REAL – WTR (4. : Maio 2003 : Natal, RN). Anais. p. 47-54.
- [5] Dibble, P.C. Real-Time Java Platform Programming. 1. ed. Prentice-Hall, 2002.
- [6] Bollella, G., Brosgol, B., Dibble, P., et al. The Real-Time Specification for Java. Revision 1.0. Addison-Wesley, 2000. Disponível em <http://www.rtyj.org> , acesso em Março, 2004.