

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ANDERSON ANTÔNIO GIACOMOLLI

**PROPOSTA DE ARQUITETURA
ORIENTADA A SERVIÇOS PARA UM
SISTEMA DE MANUTENÇÃO
INTELIGENTE**

Porto Alegre
2014

ANDERSON ANTÔNIO GIACOMOLLI

**PROPOSTA DE ARQUITETURA
ORIENTADA A SERVIÇOS PARA UM
SISTEMA DE MANUTENÇÃO
INTELIGENTE**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Controle e Automação

ORIENTADOR: Prof. Dr. Carlos Eduardo Pereira

Porto Alegre
2014

ANDERSON ANTÔNIO GIACOMOLLI

**PROPOSTA DE ARQUITETURA
ORIENTADA A SERVIÇOS PARA UM
SISTEMA DE MANUTENÇÃO
INTELIGENTE**

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Dr. Carlos Eduardo Pereira, UFRGS

Doutor pela Stuttgart University – Stuttgart, Alemanha

Banca Examinadora:

Prof. Dr. José Antônio Barata de Oliveira, UNINOVA

Doutor pela Universidade Nova de Lisboa – Lisboa, Portugal

Prof. Dr. Marcelo Götz, UFRGS

Doutor pela Universidade Paderborn – Alemanha e UFRGS, Brasil

Prof. Dr. Renato Ventura Bayan Henriques, UFRGS

Doutor pela Universidade Federal de Minas Gerais – Belo Horizonte, Brasil

Coordenador do PPGEE: _____

Prof. Dr. Arturo Suman Bretas

Porto Alegre, março de 2014.

AGRADECIMENTOS

À minha família, pelo apoio incondicional e auxílio durante o período de elaboração e conclusão deste trabalho.

Ao meu orientador, professor Carlos Eduardo Pereira, pela confiança e oportunidade de trabalho na área de pesquisa em que se enquadra este trabalho.

Aos colegas e amigos, Carlos Solon, Eduardo Maciel, Gustavo Künzel e Thiago Michelin, pelas incontáveis horas de estudo e também descontração.

Ao amigo Ronaldo Hüseemann, pelos conselhos e apoio na decisão de fazer o mestrado.

À Daniela, por todo amor, carinho e compreensão.

RESUMO

No âmbito industrial, o custo empregado na manutenção de equipamentos ainda representa uma grande parcela dos investimentos. Dessa forma, o desenvolvimento de técnicas de manutenção, e o seu correto planejamento, cada vez mais estão assumindo papéis de grande importância nesse setor, visto que impactam diretamente no fator econômico das empresas. Neste sentido, o trabalho em questão apresenta a proposta de uma arquitetura orientada a serviços para um sistema de manutenção inteligente, a fim de possibilitar a integração de forma facilitada entre equipamentos e as ferramentas de análise de degradação. A arquitetura é composta por diferentes entidades, cada uma responsável por determinada tarefa para o funcionamento do sistema. No trabalho, as entidades são implementadas e experimentos são realizados a fim de validar a solução proposta.

Palavras-chave: Sistema de Manutenção Inteligente, Arquitetura Orientada a Serviços, Integração de Sistemas, Automação Industrial.

ABSTRACT

In the industrial field, the costs associated with equipment's maintenance still represents a large portion of the financial resources available to a company. Therefore, new researches in maintenance systems, and the correct task planning, are growing, since they impact directly on the economic side of the companies. Thus, current work presents a service-oriented architecture for maintenance systems integration. The proposed architecture intends to facilitate the integration of equipments and degradation analysis tools. The architecture is comprised of several entities, where each one is responsible for executing a given task in order to keep the system running properly. In this work, all the entities are implemented and experiments are performed in order to validate the proposed system.

Keywords: Intelligent Maintenance Systems, Service-Oriented Architectures, Systems Integration, Industrial Automation.

LISTA DE ILUSTRAÇÕES

| | | |
|------------|--|----|
| Figura 1: | Classificação das estratégias de manutenção. | 27 |
| Figura 2: | Processamento das informações utilizando a estratégia proposta pelo IMS Center. | 27 |
| Figura 3: | Representação do conceito de valor de confiança – adaptado de (DJURD-JANOVIC; LEE; NI, 2003). | 28 |
| Figura 4: | Modelo OSA-CBM – adaptado de (MIMOSA, 2014). | 29 |
| Figura 5: | Representação da resolução tempo frequência da transformada wavelet. | 31 |
| Figura 6: | Representação de dois níveis da árvore de decomposição wavelet packet. | 31 |
| Figura 7: | Representação de um modelo de regressão logística. | 32 |
| Figura 8: | Interoperabilidade entre os elementos de uma aplicação SOA. | 35 |
| Figura 9: | Pilha de protocolos suportados pelo DPWS. | 36 |
| Figura 10: | Arquitetura orientada a serviços proposta para integração de um sistema de manutenção inteligente. | 46 |
| Figura 11: | Diagrama de casos de uso para descoberta e configuração de dispositivos. | 51 |
| Figura 12: | Diagrama de casos de uso para a configuração do envio dos dados do dispositivo. | 51 |
| Figura 13: | Possíveis visualizações da topologia dos dispositivos: (a) visão da rede; (b) visão lógica. | 52 |
| Figura 14: | Diagrama de casos de uso para envio de dados de treinamento para um dispositivo. | 54 |
| Figura 15: | Diagrama de casos de uso para o gerenciamento de análises pelo operador do sistema. | 55 |
| Figura 16: | Busca de dispositivo e criação de um plano de análise. | 56 |
| Figura 17: | Diagrama de casos de uso para a obtenção e execução das análises. | 57 |
| Figura 18: | Diagrama de execução de uma análise de equipamento. | 57 |
| Figura 19: | Diagrama de casos de uso para obtenção dos relatórios de saúde do dispositivo. | 58 |
| Figura 20: | Atuador elétrico modelo CSR6. | 59 |
| Figura 21: | Diagrama da localização dos sensores instalados no atuador elétrico. | 60 |
| Figura 22: | Engrenagens utilizadas no estudo de caso – adaptado de (LAZZARETTI, 2012). | 60 |
| Figura 23: | Localização das engrenagens satélite no atuador elétrico. | 61 |
| Figura 24: | Visão geral dos clientes, dispositivos e serviços hospedados. | 61 |
| Figura 25: | Tela principal do Gerenciador de Dispositivos. | 68 |

| | | |
|------------|--|----|
| Figura 26: | Diálogo com informações detalhadas sobre o dispositivo e lista de sensores disponíveis. | 69 |
| Figura 27: | Diálogo de configuração do Gerenciador de Dispositivos. | 70 |
| Figura 28: | Diálogo para configuração dos comportamentos do dispositivo. . . . | 72 |
| Figura 29: | Diagrama de classes da interface para criação de novos comportamentos. | 72 |
| Figura 30: | Diálogo de envio de dados de treinamento. | 73 |
| Figura 31: | Diálogo de definição e configuração de grupos de dispositivos. . . . | 73 |
| Figura 32: | Diálogo para obtenção de relatórios de saúde. | 74 |
| Figura 33: | Tela principal do Gerenciador de Análises. | 75 |
| Figura 34: | Diálogo de criação de uma nova análise. | 75 |
| Figura 35: | Diálogo para escolha das ferramentas utilizadas na análise. | 76 |
| Figura 36: | Diálogo para a configuração dos comportamentos dos dispositivos. . . | 77 |
| Figura 37: | Tela principal do Analisador de Dispositivos. | 77 |
| Figura 38: | Diagrama de interoperabilidade entre o Analisador de Dispositivos e os <i>scripts</i> do Watchdog Agent executando no software Matlab. | 79 |
| Figura 39: | Diagrama de classes para abstração das ferramentas de análise. | 79 |
| Figura 40: | Tela principal do software para simulação dos dispositivos. | 81 |

LISTA DE TABELAS

| | | |
|-----------|--|----|
| Tabela 1: | Metadados disponíveis para configuração em cada dispositivo. | 62 |
| Tabela 2: | Parâmetros configuráveis dos sensores. | 62 |
| Tabela 3: | Situações definidas para coleta dos dados. | 63 |

LISTA DE ABREVIATURAS

| | |
|------------|--|
| CSV | Comma-Separated Values |
| DPWS | Devices Profile for Web-Services |
| ERP | Enterprise Resource Planning |
| FPGA | Field-Programmable Gate Array |
| HTTP | Hypertext Transfer Protocol |
| IHM | Interface Homem-Máquina |
| IMS Center | Intelligent Maintenance System Center |
| JAR | Java Archive |
| JMEDS | Java Multi Edition DPWS Stack |
| MES | Manufacturing Execution Systems |
| OPC UA | OPC Unified Architecture |
| OSA-CBM | Open Systems Architecture for Condition-Based Maintenance |
| SIRENA | Service Infrastructure for Real Time Embedded Networked Applications |
| SOA | Service-Oriented Architecture |
| SOAP | Simple Object Access Protocol |
| TCP | Transmission Control Protocol |
| UDDI | Universal Description Discovery Integration |
| UML | Unified Modeling Language |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| WSDL | Web Service Description Language |
| XML | Extensible Markup Language |

LISTA DE SÍMBOLOS

| | |
|---------------|---|
| β | Varição da dilatação |
| κ | Varição da escala |
| k | Dimensão do espaço |
| r | Vetor de entrada do modelo de regressão logística |
| y | Saída do modelo de regressão logística |
| ψ | Função wavelet mãe |
| \mathcal{W} | Transformada wavelet contínua |
| \mathcal{V} | Transformada wavelet discreta |
| $x(t)$ | Sinal contínuo no domínio tempo |
| α | Parâmetro de dilatação |
| τ | Parâmetro de translação |

SUMÁRIO

| | | |
|------------|---|----|
| 1 | INTRODUÇÃO | 21 |
| 2 | CONCEITUAÇÃO TEÓRICA | 25 |
| 2.1 | Sistemas de manutenção inteligente | 25 |
| 2.1.1 | A ferramenta Watchdog Agent | 27 |
| 2.1.2 | Modelo OSA-CBM | 28 |
| 2.1.3 | Algoritmos de processamento da ferramenta Watchdog Agent | 30 |
| 2.2 | Arquiteturas orientadas a serviços | 33 |
| 2.2.1 | Componentes de uma arquitetura orientada a serviços | 33 |
| 2.2.2 | Device Profile for Web Services | 35 |
| 3 | ANÁLISE DO ESTADO DA ARTE | 39 |
| 3.1 | Sistemas de manutenção inteligente | 39 |
| 3.2 | Arquiteturas orientadas a serviços | 41 |
| 3.3 | Uso de arquiteturas orientadas a serviços em conjunto com sistemas de manutenção inteligente | 43 |
| 4 | ARQUITETURA PROPOSTA | 45 |
| 4.1 | Arquitetura orientada a serviços proposta | 45 |
| 4.1.1 | Serviço | 46 |
| 4.1.2 | Dispositivo | 46 |
| 4.1.3 | Gerenciador de Dispositivos | 47 |
| 4.1.4 | Gerenciador de Análises | 48 |
| 4.1.5 | Analisador de Dispositivos | 48 |
| 4.1.6 | Base de dados | 49 |
| 4.1.7 | Repositório de serviços | 49 |
| 4.2 | Casos de uso para a arquitetura proposta | 49 |
| 4.2.1 | Descoberta e configuração de dispositivos | 50 |
| 4.2.2 | Envio de dados de treinamento | 53 |
| 4.2.3 | Gerenciamento de análises | 54 |
| 4.2.4 | Análise dos dados do dispositivo | 56 |
| 4.2.5 | Relatórios de saúde | 57 |
| 4.3 | Estudo de caso | 58 |
| 4.3.1 | Visão geral do objeto de estudo de caso | 58 |
| 4.3.2 | Definição dos dispositivos | 60 |
| 4.3.3 | Aquisição dos dados | 62 |
| 4.3.4 | Análise de degradação | 63 |

| | | |
|------------|--|----|
| 4.4 | Definição dos experimentos | 64 |
| 4.4.1 | Verificação da interoperabilidade entre as entidades | 64 |
| 4.4.2 | Comparação da proposta com métodos tradicionais | 65 |
| 4.4.3 | Verificação da escalabilidade da proposta | 65 |
| 5 | IMPLEMENTAÇÃO E RESULTADOS | 67 |
| 5.1 | Implementação das entidades da arquitetura proposta | 67 |
| 5.1.1 | Implementação do Gerenciador de Dispositivos | 68 |
| 5.1.2 | Implementação do Gerenciador de Análises | 74 |
| 5.1.3 | Implementação do Analisador de Dispositivos | 77 |
| 5.1.4 | Implementação da camada de acesso a base de dados e definição da estrutura interna | 80 |
| 5.1.5 | Implementação dos dispositivos | 80 |
| 5.2 | Resultados obtidos com a arquitetura proposta | 82 |
| 5.2.1 | Verificação da interoperabilidade entre as entidades | 82 |
| 5.2.2 | Comparação da proposta com métodos tradicionais | 84 |
| 5.2.3 | Verificação da escalabilidade da proposta | 84 |
| 6 | CONCLUSÃO | 87 |
| | REFERÊNCIAS | 91 |
| | APÊNDICE A: CÓDIGOS UTILIZADOS NA PROPOSTA | 97 |

1 INTRODUÇÃO

Atualmente, a importância do emprego de técnicas de manutenção no âmbito industrial está em constante ascensão devido à necessidade de aumentar a disponibilidade e segurança dos equipamentos, bem como a qualidade do processo produtivo (MULLER; SUHNER; IUNG, 2008). O custo empregado anualmente com processos de manutenção está na faixa de 15% para a indústria de manufatura, entre 20% a 30% para a indústria química e na faixa de 40% para a indústria do aço e siderúrgica, como ilustrado nos trabalhos de (CHU; PROTH; WOLFF, 1998) e (NGUYEN; BRAMMER; BAGAJEWICZ, 2008). Dessa forma, o desenvolvimento de novas técnicas de manutenção para uso nas mais diversas áreas e o correto planejamento dos processos de manutenção estão cada vez mais importantes, uma vez que impactam diretamente no fator econômico, alterando a disponibilidade do sistema e também a segurança (ZHAO et al., 2010a).

Nos últimos anos, tem-se observado um crescimento no uso de um novo paradigma de manutenção denominado de manutenção inteligente (ZHANG; CAO; LEE, 2013). Este novo paradigma visa transformar a forma como as técnicas de manutenção são utilizadas. Diferentemente dos métodos tradicionais, conhecidos por consertar um determinado defeito em um equipamento somente após a falha ou por manterem processos de manutenção agendados baseado no histórico de falhas dos componentes, o paradigma de manutenção inteligente visa prever a condição de saúde ou funcionamento do sistema e prevenir uma possível falha. Segundo (BLOCH; GEITNER, 2012), 99% das falhas em sistemas mecânicos podem ser observadas por indicadores perceptíveis. Consequentemente é possível a utilização de técnicas de manutenção inteligente empregadas no monitoramento contínuo da saúde do sistema, de forma a não interromper a operação dos equipamentos.

As tecnologias empregadas na manutenção contínua do sistema, bem como diagnóstico de falhas, tiveram grande desenvolvimento nas últimas décadas e visam prever o estado do sistema (HENG et al., 2009). Assinaturas de sinais de qualquer natureza puderam ser obtidas, processadas e analisadas através de sensores e softwares computacionais. Novas pesquisas nestas áreas estão em constante evolução, utilizando técnicas modernas para análise e processamento de sinais. Isso se torna possível com os avanços da eletrônica e computação, que cada vez mais propiciam ferramentas e técnicas para a resolução

de problemas (ZHAO et al., 2010b). Como exemplo, pode-se citar o uso de métodos baseados em redes neurais ou de mapas auto-organizáveis para detecção de padrões em sinais (GONÇALVES et al., 2011).

Outra linha de pesquisa também está em constante crescimento: o uso de Service-Oriented Architecture (SOA) ou Arquiteturas Orientadas a Serviços. O uso do padrão SOA está evoluindo e está cada vez mais presente em aplicações nos mais diversos segmentos, sejam eles a nível de dispositivos, na implementação de camadas de negócios ou mesmo no setor industrial, como apresentado em (CANDIDO et al., 2010a), (CHOI; NAZARETH; JAIN, 2010), (RAGAVAN; KUSNANTO; GANAPATHY, 2012) e (PAPAZOGLU; HEUVEL, 2007). É um conceito de arquitetura que suporta acoplamento mínimo entre componentes, possibilitando ganhos em flexibilidade e interoperabilidade. Por conseguinte, qualquer tipo de aplicação pode ser representada como um conjunto complexo de serviços.

Com a utilização de SOA, um recurso ou componente é identificado como um serviço. Cada entidade apresenta comportamento bem definido e é composta por módulos autocontidos, os quais permitem que um determinado serviço seja independente do estado ou contexto de outros serviços (PAPAZOGLU; HEUVEL, 2007). As funcionalidades agregadas a um serviço são publicadas e disponibilizadas através de uma interface padrão, o que possibilita a troca de informações ou requisição da execução de alguma tarefa entre os componentes (RAGAVAN; KUSNANTO; GANAPATHY, 2012).

O uso de SOA no âmbito industrial, como forma de integração dos sistemas, se mostra factível pelos sucessos de vários projetos, dos quais pode-se citar (KARNOUSKOS et al., 2010), (BOHN; BOBEK; GOLATOWSKI, 2006), (DEUGD et al., 2006) e (COLOMBO; KARNOUSKOS; MENDES, 2010). Os projetos demonstram a viabilidade na utilização de serviços em sistemas embarcados a fim de integrá-los com sistemas Manufacturing Execution Systems (MES) e Enterprise Resource Planning (ERP), localizados nos níveis mais altos da corporação. A utilização de SOA em ambientes industriais possibilita o aumento da flexibilidade do sistema, resultando em rápida adaptação para situações onde são impostas demandas do mercado (STARKE; KUNKEL; HAHN, 2013). A configuração de equipamentos de forma flexível utilizando SOA possibilita aumento na agilidade de como os processos desta natureza são executados.

Nesse contexto, um sistema de manutenção inteligente também pode se valer da utilização dos conceitos empregados pelo padrão SOA. Do ponto de vista da arquitetura SOA, o sistema de manutenção pode conter serviços para relatórios da condição de saúde e de falhas nos equipamentos, informações sobre o prognóstico do tempo de operação sem necessidade de manutenção, além de serviços de configuração de ferramentas de diagnóstico ou dos modos de operação suportados pelo equipamento. O monitoramento remoto do sistema também possibilita a integração das informações de saúde dos equipamentos em sistemas MES e ERP, a fim de se obter o correto gerenciamento da cadeia de

suprimentos de peças de reposição (OLDHAM; JAMES; SHAW, 2003).

Mesmo que as pesquisas envolvendo as áreas de sistemas de manutenção e arquiteturas orientadas a serviços estejam em constante evolução, as iniciativas para integração das duas tecnologias ainda são escassas. Isso posto, o trabalho em questão apresenta a proposta de uma arquitetura orientada a serviços para um sistema de manutenção inteligente. A proposta tem por objetivo possibilitar a integração de equipamentos com um sistema de manutenção inteligente de forma flexível. Tanto equipamentos de campo quanto outras entidades relacionadas com o sistema de análise são definidas como componentes SOA.

A arquitetura proposta engloba diversas entidades, cada uma destinada a um propósito específico. As entidades foram construídas conforme a demanda encontrada para a integração entre os sistemas aqui descritos. Dentre as principais, estão o Analisador de Dispositivos, que possibilita a análise automática de equipamentos, e o Gerenciador de Análises, o qual permite a definição da forma como os dados dos equipamentos serão analisados, a fim de se obter um indicador do nível de degradação. Além dessas, a arquitetura também conta com ferramentas para configuração dos equipamentos a serem analisados e da integração com uma base de dados, a fim de armazenar todas as informações da análise, desde as intermediárias até o resultado de degradação.

Os planos de análise representam uma parte importante deste trabalho. Armazenam as informações relativas à análise de determinado equipamento, como as ferramentas necessárias para o cálculo dos níveis de degradação e as possíveis ações a serem tomadas quando determinado nível for obtido. As ferramentas de análise se referem aos algoritmos de processamento e extração de características dos sinais provenientes dos sensores dos equipamentos. Os planos são definidos pelo operador do sistema e executam automaticamente pela entidade de análise de equipamentos.

Dentre as diversas tecnologias disponíveis para a integração dos elementos da arquitetura proposta SOA, optou-se pela especificação Devices Profile for Web-Services (DPWS). A especificação DPWS emprega um conjunto restrito de padrões para serviços web que pode ser utilizado também em sistemas embarcados. Dessa forma, pela natureza da aplicação proposta neste trabalho, que inclui a comunicação entre equipamentos de campo e entidades de mais alto nível de processamento, DPWS se torna uma escolha viável. Esta escolha é comprovada no projeto e implementação dos dispositivos.

Como objetivos do trabalho, estão a definição da arquitetura proposta e implementação das entidades. A validação do sistema é feita com a implementação de um dispositivo que representa um conjunto atuador elétrico e válvula, utilizado para controle de fluxo em redes de distribuição de petróleo. Com a definição do dispositivo que representa o atuador elétrico, o sistema é posto em funcionamento, onde o operador do sistema tem acesso às configurações dos equipamentos e planos de análise, podendo criar novos planos e verificar o resultado dos níveis de degradação obtidos com as análises.

Por fim, são ilustrados experimentos para determinar a viabilidade da solução proposta. Os experimentos tem por objetivo verificar a interoperabilidade entre as entidades, bem como determinar o correto funcionamento dos componentes implementados. São analisados pontos positivos em relação à utilização do sistema proposto, tendo em vista a comparação com a utilização de métodos tradicionais de obtenção dos níveis de degradação em equipamentos. Também são verificadas as vantagens na utilização da arquitetura proposta em relação ao aumento do número de dispositivos monitorados, levando em conta o gerenciamento de vários dispositivos similares para a obtenção dos estados de saúde de todos.

Esta dissertação está estruturada da seguinte forma: no capítulo 2 são apresentados os conteúdos teóricos utilizados ao longo do trabalho, divididos nas duas áreas de estudo, sendo elas a de manutenção inteligente e também arquiteturas orientadas a serviços; a análise do estado da arte é apresentada no capítulo 3, onde são descritos os estudos atuais envolvendo os dois assuntos pertinentes a este trabalho; a proposta de uma arquitetura orientada a serviços faz parte do capítulo 4, onde é apresentado o projeto dos componentes utilizados no sistema proposto, além da definição do estudo de caso e dos experimentos para validação; no capítulo 5 são apresentados os resultados obtidos com a implementação do sistema proposto e o resultado dos experimentos definidos no capítulo anterior; por fim, o capítulo 6 apresenta as conclusões e pontos relevantes para a continuidade do trabalho.

2 CONCEITUAÇÃO TEÓRICA

Este capítulo apresenta os conceitos básicos utilizados ao longo do desenvolvimento do trabalho. Basicamente são discutidas as duas áreas principais que envolvem o estudo proposto: sistemas de manutenção inteligente e arquiteturas orientadas a serviços. Em se tratando de manutenção inteligente, são discutidos os tipos de manutenção empregados e as novas tendências, que envolvem a utilização de técnicas de predição e monitoramento contínuo do sistema. Também é apresentado o software Watchdog Agent e as ferramentas utilizadas para manipulação dos dados.

Na conceituação teórica de arquiteturas orientadas a serviços, são explanadas as principais características deste tipo de abordagem. Além disso, são apontados os elementos principais que fazem parte do sistema e implementações disponíveis. Uma abordagem mais detalhada é ilustrada para a utilização de DPWS, que é a tecnologia utilizada neste trabalho para a construção da aplicação SOA.

2.1 Sistemas de manutenção inteligente

Manutenção, no âmbito geral, consiste em uma série de medidas de prevenção, correção e predição de falhas (LEE et al., 2006). Durante o uso, equipamentos ou máquinas tendem a deteriorar e alterar o seu padrão de funcionamento devido a diversos fatores, como, por exemplo, desgaste, rachaduras, corrosão e sujeira. Nestas condições, a restauração do sistema é de suma importância, visto que, com o passar do tempo, podem apresentar defeitos e levar a falhas e indisponibilidades. De acordo com (MARÇAL; SUSIN, 2005), manutenção pode ser definida como todas as atividades técnicas e organizacionais que garantam a operação das máquinas e equipamentos dentro da confiabilidade esperada.

Tradicionalmente, são encontrados na literatura três tipos de estratégias de manutenção: corretiva, preventiva e preditiva (MARÇAL; SUSIN, 2005). A manutenção corretiva visa reestabelecer os sistemas danificados; a preventiva tem por objetivo manter os sistemas em funcionamento, realizando pequenas correções; e a manutenção preditiva tem por base o monitoramento do estado do sistema, detectando falhas insipientes e fornecendo

subsídios para o planejamento de ações de prevenção ou correção. Em termos gerais, a manutenção corretiva é aplicada somente quando há falha e o sistema necessita de reposição de peças ou componentes para continuar operando corretamente, enquanto que a manutenção preventiva visa o agendamento programado de intervenções no sistema, afim de manter o funcionamento pelo maior tempo possível. Por outro lado, a manutenção preditiva tem como foco o monitoramento do sistema continuamente e, desta forma, a intervenção é feita somente quando necessário.

Nas três estratégias, pode-se citar vantagens e desvantagens. Na manutenção corretiva, a principal vantagem está na dispensabilidade de realização de acompanhamentos ou inspeções no sistema. Isso evita a geração de custos na alocação de pessoas ou equipamentos para desempenharem tarefas de verificação do sistema, além da parada da linha de produção em intervalos agendados. Por outro lado, a parada inesperada da linha de produção para uma manutenção emergencial pode gerar transtornos e custos não programados. A manutenção preventiva visa sanar os problemas de paradas inesperadas, utilizando-se de um modelo de agendamento das inspeções. O que para muitas situações é considerado suficiente, se não for bem planejado, usualmente acarreta em custos excessivos devido às paradas programadas e alocação de equipes de manutenção. Nesta situação, a estratégia de manutenção preditiva busca a otimização do número de paradas inesperadas, utilizando-se da predição do estado do sistema, como análise de tendências ou avaliações probabilísticas do estado de degradação dos equipamentos, para os agendamentos de novas intervenções.

Mesmo com a programação das intervenções, as máquinas podem falhar de modo repentino, pondo em risco os equipamentos e pessoas envolvidas com o processo produtivo (GONÇALVES, 2011). A falha no intervalo entre intervenções não é possível de prever através dos métodos clássicos de manutenção. Logo, nos últimos anos, o que tem se visto é a substituição da estratégia de manutenção preventiva por um novo paradigma: a manutenção proativa (LEE et al., 2009). Esta nova estratégia visa não somente a predição do estado do sistema, mas também o diagnóstico das falhas e, em casos onde é aplicado, a intervenção de forma automática. Por intervenção, entende-se que o padrão de funcionamento dos equipamentos monitorados pode ser alterado, visando minimizar os possíveis agravantes até a realização da manutenção.

Neste cenário, emergem os sistemas de manutenção inteligente. Também conhecidos como sistemas de manutenção baseados no conhecimento, visam capturar o conhecimento de um determinado sistema sob a forma de regras e utilizá-las para construir um novo sistema baseado nestas regras. O novo sistema é, então, utilizado para realização de um correto diagnóstico ou tomada de ação no caso da ocorrência de algum defeito. Como exemplo, em (SHIKARI; SADIWALA; DWIVEDI, 2004) o padrão de vibração de uma máquina de indução, de um atuador e de uma prensa são analisados e, realizado o diagnóstico automático através de um sistema de manutenção inteligente, é determinado

o motivo da falha, podendo ser os rolamentos ou desalinhamentos. As três estratégias clássicas de manutenção juntamente com a de manutenção inteligente são apresentadas na figura 1.

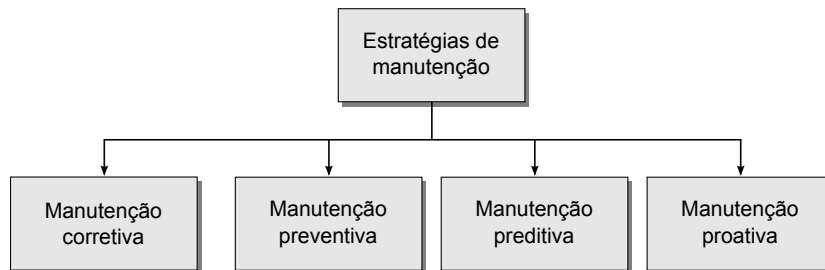


Figura 1: Classificação das estratégias de manutenção.

Com o intuito de auxiliar na migração do paradigma de conserto após falha para o paradigma de predição e prevenção, foi criado, nos Estados Unidos, um centro composto por universidades e empresas, denominado Intelligent Maintenance System Center (IMS Center). Dentre as empresas integrantes da parceria IMS Center, pode-se citar, por exemplo, Boeing, Siemens, AMD, Toyota e Caterpillar. Entre as universidades, fazem parte do consórcio a de Cincinnati, Missouri-Rolla e Michigan (IMS CENTER, 2014).

2.1.1 A ferramenta Watchdog Agent

Um dos objetivos da parceria IMS Center foi o desenvolvimento de uma metodologia para abordagem dos problemas de manutenção utilizando predição e prevenção. Para tanto, foi desenvolvido um conjunto de ferramentas de análise denominado Watchdog Agent. Em termos gerais, o Watchdog Agent é uma ferramenta de análise de desempenho. Aplicado a determinado equipamento, visa analisar sinais de diversas partes da máquina, a fim de obter um índice de desempenho.

A extração das informações contidas nos sinais analisados são extraídas por meio das ferramentas implementadas no Watchdog Agent. Primeiramente os dados dos sensores do equipamento são adquiridos. Em um segundo momento, os dados são classificados com o auxílio de algoritmos. Com os dados classificados, é possível determinar o índice de desempenho para a situação analisada. Estas etapas são ilustradas na figura 2.



Figura 2: Processamento das informações utilizando a estratégia proposta pelo IMS Center.

À medida que o equipamento degrada, o índice de desempenho é alterado em comparação com o mesmo indicador obtido com o equipamento em funcionamento normal. Um indicador normalmente utilizado para identificação do estado de um equipamento é o valor de confiança. Este indicador é definido como uma grandeza que varia no intervalo $[0; 1]$. Em determinadas condições, pode-se assumir que valores próximos a 1 representam funcionamento normal do sistema, enquanto que valores próximos a 0 equivalem a um funcionamento em falha. A figura 3 ilustra o conceito de valor de confiança. As duas curvas da esquerda apresentam o comportamento normal e o comportamento recente de um determinado equipamento. Ao cruzar as duas informações, é possível obter o valor de confiança, exemplificado no gráfico da direita. À medida que o valor de confiança decai, a probabilidade de redução do desempenho do sistema aumenta (DJURDJANOVIC; LEE; NI, 2003).

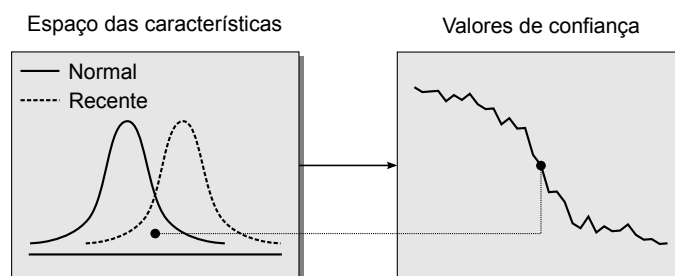


Figura 3: Representação do conceito de valor de confiança – adaptado de (DJURDJANOVIC; LEE; NI, 2003).

Em comparação com estratégias de manutenção preventiva, um ponto importante a ser citado é aumento da vida útil de peças de equipamentos (LAZZARETTI, 2012). Peças que poderiam ser descartadas em função de uma intervenção preventiva podem ser utilizadas por um período maior de tempo, até que, devido a desgastes ou outros fatores, a redução no desempenho do sistema seja comprometida.

2.1.2 Modelo OSA-CBM

Como proposta de padronização de uma arquitetura aberta para troca de informações em um sistema baseado em condição, surge o modelo Open Systems Architecture for Condition-Based Maintenance (OSA-CBM) (THURSTON, 2001). A arquitetura OSA-CBM visa facilitar a integração e interoperabilidade entre componentes e equipamentos de diferentes fabricantes. Definida em sete camadas, possibilita a abstração de várias partes envolvidas em um sistema de manutenção. A figura 4 apresenta uma visão geral das camadas do modelo juntamente com as suas interações. As camadas são numeradas de 1 (aquisição de dados) a 7 (apresentação).

A definição das funcionalidades de cada camada é apresentada por (THURSTON, 2001). Na camada de aquisição de dados, as grandezas físicas são convertidas para sinais elétricos e digitalizadas. O módulo consiste, normalmente, de um elemento sensor e um

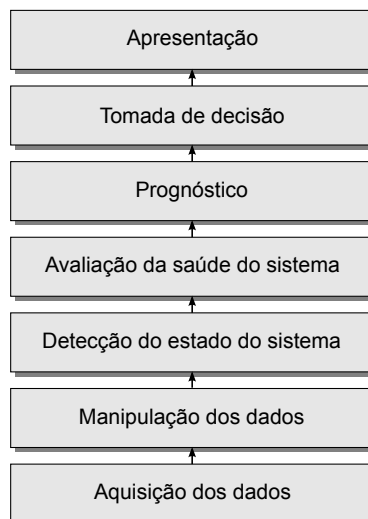


Figura 4: Modelo OSA-CBM – adaptado de (MIMOSA, 2014).

elemento de aquisição de dados. Além da conversão física, a camada também pode armazenar os dados coletados em um banco de dados. A primeira etapa de cálculos sob os dados obtidos é feita na camada de manipulação dos dados. Através do uso de ferramentas de processamento de sinais, os dados adquiridos na camada anterior são manipulados, podendo gerar resultados no domínio tempo, frequência ou tempo-frequência. Eventualmente, os resultados das operações também podem ser armazenados em um banco de dados. O módulo de detecção do estado do sistema analisa continuamente os indicadores de cada sistema, subsistema ou componente. De posse dos dados processados pelas camadas anteriores, ao calcular os indicadores de estado, o módulo de detecção pode gerar alarmes respeitando condições previamente estabelecidas. Novamente os dados obtidos podem ser armazenados para uso posterior. Na camada de avaliação da saúde do sistema, o resultado dos indicadores, obtidos no módulo de detecção do estado do sistema, são inseridos no contexto das operações. A saúde do sistema monitorado é avaliada pela comparação dos indicadores de saúde passados e atuais, respectivamente. Dessa forma, também é possível armazenar os resultados formando um histórico do equipamento monitorado. Na camada de prognóstico, a saúde futura do sistema é estimada. Através de um modelo estimado do sistema e dos dados obtidos nas camadas anteriores, o tempo de vida útil ou a probabilidade de falha em um horizonte de predição são estimados. Como nas camadas anteriores, os resultados podem ser armazenados em um banco de dados. O módulo de tomada de decisão utiliza os dados obtidos na camada de prognóstico, além de outras informações, para sugerir ações recomendadas de acordo com as implicações das decisões. São integrados, juntamente com os dados da camada de prognóstico, informações de restrições externas, requisitos de funcionalidades do equipamento ou sistema, condições financeiras, entre outros. No nível mais alto do modelo, está a camada de aplicação. Definida como a interface homem-máquina do sistema, visa a apresentação dos dados obtidos no processamento das informações. Nesta camada também podem ser uti-

lizadas técnicas de realidade aumentada como, por exemplo, proposto em (ESPÍNDOLA, 2011).

2.1.3 Algoritmos de processamento da ferramenta Watchdog Agent

Como mencionado anteriormente, o Watchdog Agent é um conjunto de algoritmos para processamento de sinais e extração de características. A ferramenta é dividida em módulos seguindo o modelo OSA-CBM e possui implementações nas camadas de manipulação dos dados, detecção do estado e avaliação da saúde dos sistema. As ferramentas descritas a seguir fazem referência às utilizadas neste trabalho.

2.1.3.1 Energias da transformada Wavelet Packet

Uma das forma de se analisar sinais não estacionários no domínio tempo-frequência é utilizando a transformada wavelet (ANTONINI et al., 1992). Seu uso é indicado para sinais que apresentam descontinuidades, tendências entre outros, sendo empregada nas mais diversas aplicações, desde a remoção de ruídos em sinais ou imagens até a compressão de imagens médicas com pouca perda de qualidade.

Wavelets são formas de onda oscilantes com duração limitada e valor médio zero. São empregadas na forma de wavelets mãe, definidas por ψ . A função wavelet mãe pode ser dilatada ou comprimida através de um parâmetro α e transladada através de um parâmetro τ . A mudança de escala e translação são apresentadas na equação (1).

$$\psi_{\alpha,\tau}(t) = \frac{1}{\sqrt{\alpha}} \psi\left(\frac{t-\tau}{\alpha}\right) \quad (1)$$

A equação (2) apresenta a definição da transformada wavelet contínua \mathcal{W} de um sinal contínuo $x(t)$.

$$\mathcal{W}\{x, \psi\} = \frac{1}{\sqrt{\alpha}} \{x(t), \psi_{\alpha,\tau}(t)\} = \frac{1}{\sqrt{\alpha}} \int_{-\infty}^{\infty} x(t) \cdot \psi^*\left(\frac{t-\tau}{\alpha}\right) dt \quad (2)$$

A largura da wavelet é influenciada pelo fator de escala α , o que também contribui para a alteração da resolução empregada na análise. Quanto menor o valor de α , maior será a resolução empregada na detecção de eventos de alta frequência. No caso contrário, quanto maior for o valor de α , maior será a dilatação empregada na wavelet mãe, o que é conveniente para a identificação de padrões de baixa frequência. A figura 5 ilustra a mudança de escala para análise do sinal em multiresolução.

Como alternativa para a utilização da transformada para sinais discretos, é possível a utilização da transformada wavelet discreta. Como a transformada wavelet contínua requer um esforço computacional considerado exagerado para calcular os coeficientes de todas as possíveis escalas da transformada, gerando informações redundantes, é possível a utilização de parâmetros de escalonamento e translação discretos (MALLAT, 1989). A

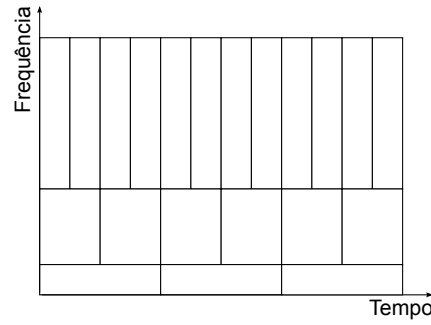


Figura 5: Representação da resolução tempo frequência da transformada wavelet.

transformada wavelet discreta \mathcal{W} de um sinal contínuo $x(t)$ é apresentada na equação (3).

$$\mathcal{W}\{x, \psi\} = \{x(t), \psi_{\kappa\beta}\} = \int_{-\infty}^{\infty} x(t) \cdot \psi_{\kappa\beta}(t) dt \quad (3)$$

A função $\psi_{\kappa\beta}$ é a wavelet mãe criada a partir de parâmetros de escala e translação discretos. A equação (4) apresenta a obtenção da função $\psi_{\kappa\beta}$, onde α é a variação da escala e τ indica a translação; β e κ são constantes discretas que indicam, respectivamente, a variação da escala e dilatação.

$$\psi_{\kappa\beta}(t) = \frac{1}{\sqrt{\alpha^\kappa}} \psi\left(\frac{t - \beta \alpha^\kappa \tau}{\alpha^\kappa}\right) \quad (4)$$

Em termos gerais, a transformada wavelet é realizada através de um processo de filtragem de vários níveis, onde cada nível apresenta um filtro em quadratura. O sinal decomposto em cada um dos níveis apresenta duas informações, definidas como *detalhe* (alta frequência) e *aproximação* (baixa frequência). A decomposição em vários níveis origina uma árvore, denominada de árvore de decomposição wavelet packet (MALLAT, 1989). Este processo é ilustrado na figura 6.

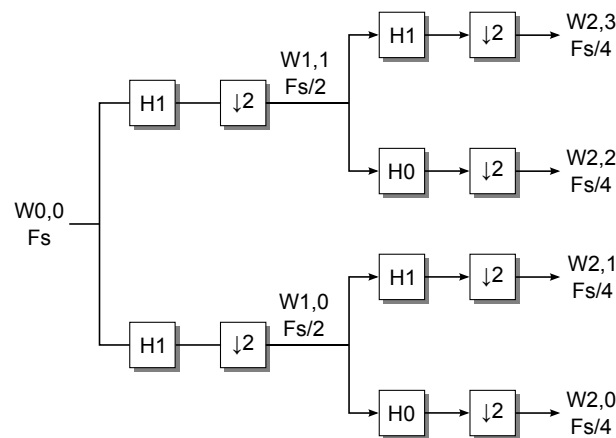


Figura 6: Representação de dois níveis da árvore de decomposição wavelet packet.

Como resultado do processo da árvore de decomposição, é gerado um vetor de elementos obtido através do cálculo da energia dos coeficientes no nível mais baixo da de-

composição da transformada. Este vetor de características é denominado de energias da transformada wavelet packet (IMS CENTER, 2007).

2.1.3.2 Regressão logística

A regressão logística é um método de classificação que permite uma classificação binária ou dicotômica de um conjunto de dados (HOSMER; LEMESHOW; STURDIVANT, 2013). O método integra a categoria de modelos chamados de Modelos Generalizados Lineares, e, portanto, como resultado da análise, é gerada uma resposta de dois estados, que podem ser traduzidos, por exemplo, para sucesso ou falha ou comportamento normal ou degradado.

O processo empregado pelo método da regressão logística é definido como a tentativa de ajustar o espaço de k dimensões da entrada para um espaço de saída de apenas uma dimensão. A variável de saída ou resposta é representada por y , sendo que $y = 1$ quando o conjunto de entrada possui a característica de interesse e $y = 0$ quando não possui (HOSMER; LEMESHOW; STURDIVANT, 2013). A figura 7 apresenta uma curva típica da saída de um modelo de regressão logística.

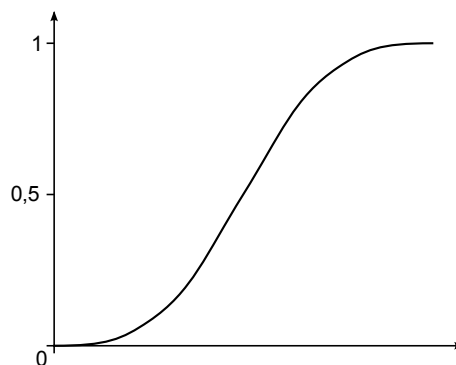


Figura 7: Representação de um modelo de regressão logística.

A representação matemática do modelo de regressão logística é apresentada na equação (5) (IMS CENTER, 2007). No modelo, r é o vetor de entrada de k dimensões e y é a saída binária.

$$p(r) = P(y = 1|r) = \frac{1}{1 + e^{-(\alpha + \beta_1 r_1 + \beta_2 r_2 + \dots + \beta_k r_k)}} = \frac{e^{\alpha + \beta_1 r_1 + \beta_2 r_2 + \dots + \beta_k r_k}}{1 + e^{\alpha + \beta_1 r_1 + \beta_2 r_2 + \dots + \beta_k r_k}} \quad (5)$$

O modelo apresentado na equação (5) também pode ser representado em termos das probabilidade de evento e de não evento. Neste caso, são definidos como $p(r)$ e $1 - p(r)$. A equação (6) apresenta a nova função, onde o termo contendo o logaritmo natural é

conhecido como função *logit*, cujo propósito é tornar a função linear.

$$g(r) = \ln \left(\frac{p(r)}{1-p(r)} \right) = \alpha + \beta_1 r_1 + \beta_2 r_2 + \dots + \beta_k + r_k \quad (6)$$

A partir da equação (5) o valor de confiança é obtido. Para um comportamento normal, o valor de confiança assume valores próximos a 1, enquanto que para comportamento ditos de falha, o valor de confiança fica concentrado próximo a 0.

2.2 Arquiteturas orientadas a serviços

As técnicas para desenvolvimento de aplicações SOA representam uma mudança de paradigma na engenharia de software, onde os componentes são definidos como serviços (RAMOLLARI; DRANIDIS; SIMONS, 2007). Originalmente desenvolvido e utilizado para integração de sistemas no meio gerencial e corporativo, logo teve aceitação entre diversos segmentos, como plataformas de negócio, telecomunicações, transportes e na automação industrial.

O termo SOA ainda não possui uma definição concisa e única, diferindo conforme os conhecimentos técnicos e a bagagem acumulada durante o desenvolvimento de diferentes aplicações por parte dos autores (CANDIDO, 2013). Ainda segundo (CANDIDO, 2013), a definição que mais se encaixa no contexto de um trabalho que envolve integração em meio industrial é a de (JAMMES; SMIT, 2005), o qual expressa que "SOA é um conjunto de princípios ou doutrinas para a construção de sistemas interoperáveis e também autônomos". Esta percepção também descreve o contexto deste trabalho, no qual os componentes envolvidos podem ser considerados peças independentes do sistema, no entanto podem vir a representar um conjunto interoperável de entidades, compartilhando recursos entre si.

Em termos gerais, a característica principal de uma arquitetura SOA é a criação e disponibilidade de serviços, que, quando agrupados, constituem um sistema funcional (JO-SUTTIS, 2009). O termo serviço se refere a uma funcionalidade ou lógica que é encapsulada e oferecida ao sistema através de uma interface. Dessa forma, outros serviços, entidades ou programas podem obter o modo de acesso à esta funcionalidade e empregá-la na resolução de determinada atividade.

2.2.1 Componentes de uma arquitetura orientada a serviços

Em se tratando do contexto da aplicação, os serviços oferecidos, para que possam ser utilizados, precisam ser encontrados ou expostos (PAPAZOGLU; HEUVEL, 2007). Mesmo que, segundo a definição adotada, serviços possam ser utilizados independentemente, a abordagem de utilizar um conjunto de serviços trabalhando de forma cooperativa na resolução de um problema é muito mais interessante. Portanto, uma aplicação SOA

deve prover meios para que os serviços possam ser comunicar e trocar informações via mensagens padronizadas. Dessa forma, é necessário que existam alguns conceitos a serem cumpridos por todos os componentes de uma arquitetura SOA (ERL, 2005):

- *Acoplamento mínimo*: serviços devem minimizar a dependência, armazenando somente as informações de outros serviços.
- *Contrato de serviço*: devem utilizar um padrão de comunicação comum previamente definido e baseado em padrões abertos.
- *Autonomia*: possibilidade de controle total da lógica que o serviço encapsula.
- *Abstração*: possibilidade de esconder a lógica e os recursos utilizados pelo serviço do resto da aplicação.
- *Reusabilidade*: utilização da mesma funcionalidade por diferentes partes do sistema ou em aplicações futuras, sem a necessidade de uma nova implementação.
- *Composição*: organização de serviços para a construção de tarefas mais complexas podem ser feitas através da composição de serviços mais simples.
- *Independência de estado*: os serviços não devem reter nenhuma informação específica sobre as atividade executadas.
- *Possibilidade de descoberta*: os serviços devem possibilitar a sua descoberta pelos mecanismos de busca.

As aplicações SOA normalmente são construídas baseadas nos princípios de serviços web (JOSUTTIS, 2009). Não necessariamente as aplicações SOA necessitam ser baseadas em serviços web, porém, esta tecnologia começou a ser largamente utilizada devido, entre outros fatores, a adoção de uma padronização. Por parte dos desenvolvedores, o encapsulamento das funcionalidades que um serviço pode oferecer foi facilitado após a definição de interfaces e protocolos de comunicação padrão. Dessa forma, é possível aos clientes acessar os serviços de forma transparente, sem conhecimento prévio de detalhes de implementação.

Seguindo as convenções adotadas para serviços web, um serviço é uma entidade de software identificada por uma Uniform Resource Identifier (URI) (BELL, 2008). A URI define o endereço do serviço na aplicação, devendo ser única. O identificador permite a discriminação entre grupos de entidades, através da utilização de um separador. Esta técnica é largamente utilizada em sistemas que usufruem de identificadores baseados em URI.

A troca de informações entre os serviços normalmente ocorre utilizando o protocolo Simple Object Access Protocol (SOAP), onde as mensagens são codificadas no formato Extensible Markup Language (XML) (JOSUTTIS, 2009). O protocolo SOAP provê uma infraestrutura básica para a troca de mensagens entre serviços web. É definido por um envelope, um conjunto de regras que definem os tipos de dados suportados e um meio de representar os procedimentos ou funcionalidades disponíveis para execução. Por ser ba-

seado em XML, o protocolo pode ser utilizado sobre diferentes protocolos de transporte, como, por exemplo, o Hypertext Transfer Protocol (HTTP).

A descrição das interfaces de cada serviço é normalmente definida por Web Service Description Language (WSDL). Documentos WSDL também são baseados em XML e contém toda a informação necessária para a utilização do serviço em questão. No documento são especificadas todas as operações que o serviço possibilita, bem como os tipos de dados suportados. Também é possível estender o documento definindo novos tipos de dados para troca de mensagens.

Os serviços disponíveis na aplicação e a descrição de suas funcionalidades estão centralizadas no Universal Description Discovery Integration (UDDI). O padrão UDDI define o protocolo para os serviços de diretório, ou intermediadores de serviço, onde são armazenadas todas as informações de cada um dos serviços da aplicação. Esta entidade é utilizada para informar aos clientes quais serviços estão disponíveis, possibilitando meios de descobri-los e obter seus metadados. A interoperabilidade entre os componentes de uma aplicação SOA é ilustrada na figura 8.

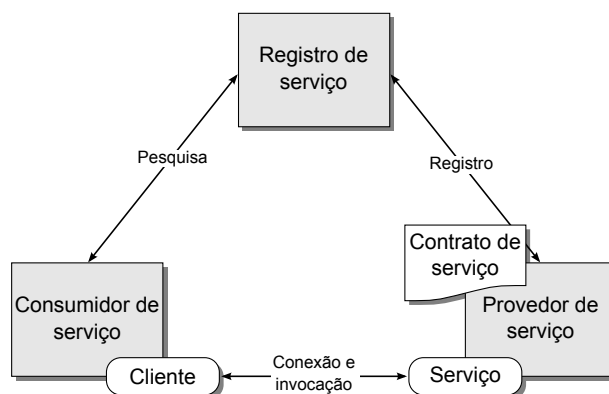


Figura 8: Interoperabilidade entre os elementos de uma aplicação SOA.

2.2.2 Device Profile for Web Services

Especificações para web services normalmente são grafadas com o prefixo "WS-" (CANDIDO, 2013). É comum encontrar na literatura o termo "WS-*", o qual se refere ao agrupamento de diferentes especificações para web services. Dentre as especificações, o DPWS define um conjunto mínimo de implementações que permitem a troca de mensagens, descoberta, descrição, geração de eventos e autenticação para a utilização de web services em clientes com recursos computacionais limitados. O DPWS permite a integração destes clientes com outros, com recursos mais flexíveis.

O DPWS implementa um conjunto restrito dos padrões WS-*. Dentre as especificações utilizadas, estão definidos formatos para o envio de mensagens de descoberta, descrição e notificação de eventos. Dos padrões, é possível destacar o WS-Addressing, utilizado para transferência de mensagens, WS-Discovery, que possibilita a descoberta de

serviços em uma rede local, e WS-Eventing, para utilização de eventos. Além das especificações WS-*, o DPWS utiliza os protocolos Transmission Control Protocol (TCP) e HTTP para a transmissão de dados. A figura 9 ilustra o diagrama contendo a pilha de protocolos suportados pelo DPWS.

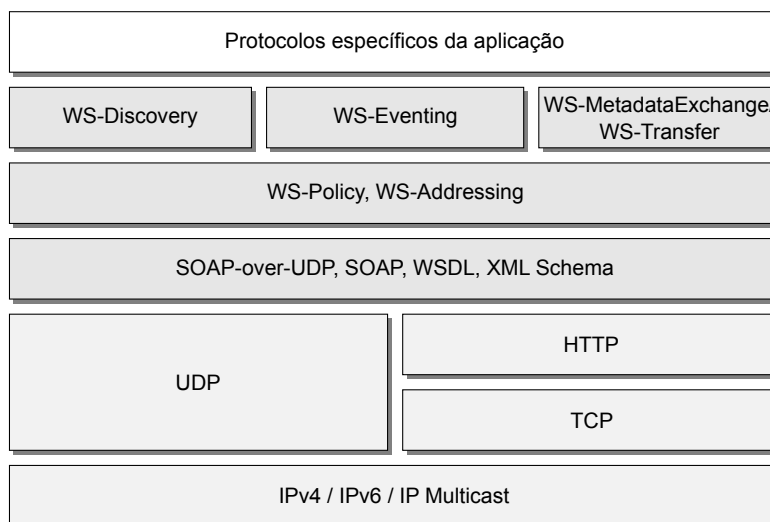


Figura 9: Pilha de protocolos suportados pelo DPWS.

O anúncio de entrada de novos dispositivos na rede é feito com o uso do WS-Discovery. Cada dispositivo que ingressa na rede pode anunciar a sua entrada através de mensagens de *Hello*. Da mesma forma, também é possível informar aos clientes a saída do dispositivo utilizando uma mensagem de *Bye*. Com a especificação WS-Discovery os clientes também podem realizar a descoberta de dispositivos na rede. Após a descoberta de um dispositivo de interesse, o cliente pode recuperar os endereços dos serviços suportados. O DPWS permite que qualquer dispositivo na rede possa enviar mensagens para a descoberta de novos dispositivos.

Um identificador único (UDDI) para cada entidade presente em uma rede DPWS, dispositivos ou serviços, é fornecido pela especificação WS-Addressing. Os identificadores únicos são denominados de *endpoints*. Através desta especificação, é possível a utilização de diferentes protocolos de transporte sem que haja necessidade de alteração na implementação do web service.

O modo como as informações de cada serviço são fornecidas aos clientes é definido através das especificações WS-MetadataExchange e WS-Transfer. As especificações permitem ao cliente buscar os metadados dos serviços disponibilizados pelos *endpoints*, como a descrição dos serviços hospedados e hospedeiros. Dentre as informações de metadados disponíveis estão o *FriendlyName*, *SerialNumber*, *FirmwareVersion*, *Manufacturer*, *ManufacturerURL*, *ModelName*, *ModelNumber*, *ModelURL* e *PresentationURL*.

A troca de informações entre dispositivos e clientes pode ser realizada através de eventos. O DPWS proporciona esta funcionalidade utilizando a especificação WS-Eventing.

O mecanismo do tipo produtor/consumidor permite que cada cliente se inscreva em eventos de interesse providos por dispositivos distintos. Após a inscrição em determinado evento, o cliente é notificado a cada alteração por parte do dispositivo. Cada inscrição, por parte do cliente, possui um tempo definido para expirar. É de responsabilidade do cliente renová-la antes deste período, a fim de continuar a receber as notificações do evento.

3 ANÁLISE DO ESTADO DA ARTE

Os trabalhos relacionados com o tema desta dissertação dizem respeito às áreas de sistemas de manutenção inteligente e arquiteturas orientadas a serviços. Neste capítulo são analisados os trabalhos relevantes para as duas áreas distintas, evidenciando os pontos favoráveis de cada um. Por fim, são apresentados trabalhos convergentes, que relacionam as duas áreas envolvidas. Os trabalhos mostram que é possível a integração entre sistemas de manutenção inteligente e arquiteturas orientadas a serviços.

3.1 Sistemas de manutenção inteligente

As ações corretivas empregadas em manutenção sempre levam à parada da linha de produção, ocasionando perdas (CARVAJAL, 2011). Estas perdas podem ser programadas ou não programadas. Em ambos os casos, existem meios de diminuir os impactos causados por estas intervenções. Ao longo dos anos, vários trabalhos comprovam o sucesso de técnicas de manutenção inteligente aplicadas em diversos cenários (MULLER; CRESPO MARQUEZ; IUNG, 2008). Entre eles, o ponto em comum está no fato do crescimento da importância do uso destas técnicas, tendo em vista a garantia da disponibilidade e segurança do sistema mantendo a qualidade da linha de produção.

O trabalho de (LEE et al., 2006) introduz os conceitos de *e-maintenance*, emergentes à época, e os elementos críticos que o compõem. Considerando que o mercado global está mais competitivo a cada ano, os autores afirmam que o uso de técnicas que minimizem os custos com a produção é de fundamental importância. Entre os custos citados, estão em destaque os que dizem respeito às falhas e quebras não programadas de maquinário. Também é sugerida a mudança de paradigma, da tradicional *falha e conserto* para *predição e manutenção*. Para contornar estes problemas, o artigo sugere o uso de técnicas computacionais, que, juntamente com o advento da Internet, possibilitam o monitoramento da condição dos equipamentos, ao invés de somente detectar os equipamentos em falha.

As ferramentas utilizadas para o monitoramento da condição dos equipamentos também são discutidas no artigo de (LEE et al., 2006). Ferramentas de avaliação da condição e predição de falhas são analisadas, a fim de se obter um monitoramento contínuo

do equipamento. Além disso, são apresentados os últimos avanços em relação às ferramentas utilizadas para este tipo de cenário e estudos de caso para validação das técnicas empregadas.

Dando continuidade aos trabalhos na área de manutenção, o artigo de (MULLER; CRESPO MARQUEZ; IUNG, 2008) aponta que a importância da manutenção em sistemas, a fim de aumentar a disponibilidade e segurança, bem como a qualidade dos produtos, está em vias de crescimento. Como o artigo anterior, o trabalho também ilustra os conceitos de *e-maintenance*, provendo uma visão geral sobre os estudos e desafios presentes neste campo de estudo. São apresentados os diferentes sistemas de manutenção disponíveis, além de uma comparação entre as diferentes características de cada um. Como pontos importantes, os autores veem grande potencial na adoção de novas tecnologias que auxiliem nas tarefas de manutenção, os quais são definidos em termos de *dispositivos inteligentes*. Além disso, defendem a criação de padrões internacionais para integração entre os sistemas discutidos. E, como contribuição maior, está a proposta da organização das ações no setor, em prol do avanço dos estudos e definição de novos conceitos, apoiando o surgimento de uma nova disciplina.

No trabalho de (GONÇALVES et al., 2009), a detecção de falhas em um atuador eletromecânico é feita através de um sistema composto por um microprocessador Microblaze sintetizado em um Field-Programmable Gate Array (FPGA). No FPGA, os sinais são processados e mapas auto-organizáveis são utilizados para detecção, classificação e predição de falhas. O treinamento dos algoritmos é realizado em um computador pessoal, enquanto que o monitoramento dos sinais é feito no FPGA. O trabalho é estendido em (GONÇALVES, 2011) com a implementação de um filtro adaptativo no FPGA. A expansão é em função da comparação das técnicas utilizadas para embarque na plataforma proposta, onde são determinados a eficiência da identificação das falhas, a área de ocupação do FPGA e o tempo de execução dos algoritmos.

O artigo de (HU; QIN, 2012) utiliza métodos baseados em redes imunológicas artificiais para prever a falha e o estado de saúde do sistema de propulsão de um navio. Considerando que, para o correto funcionamento do navio por grandes períodos de tempo, cada componente precisa estar condicionado ao correto funcionamento, os autores propõem um sistema de monitoramento para as diferentes partes do sistema de propulsão. Conforme o artigo, em navios, para aumentar a confiabilidade do sistema, são empregados muitos equipamentos redundantes, aumentando a complexidade e dificultando a identificação de uma possível falha. Nessas circunstâncias, uma falha pode acarretar em catástrofe. Com a proposta dos autores, o sistema de prognóstico ajuda a identificação das possíveis falhas em meio a complexidade imposta pelo sistema.

A proposta de uma arquitetura de manutenção inteligente baseada em agentes móveis para monitoramento das condições dos equipamentos através de um sistema que imita o sistema imunológico humano também é apresentada em (HUA; GONDAL; YAQUB,

2013). A técnica de sistemas imunológicos artificiais foi escolhida pelo fato de que muitos processos industriais, como mineração, óleo e gás, estão geograficamente localizados em áreas remotas e de difícil acesso. Baseado no fato de que alocar equipes de profissionais para inspeção destes processos demanda tempo e custo elevados, os autores propõem uma arquitetura descentralizada para automatizar o processo de monitoramento das condições dos equipamentos. O resultado dos experimentos visam determinar o desempenho do sistema proposto em termos da acurácia na detecção de falhas e alocação de banda de comunicação. Os autores avaliam que o sistema proposto não provê somente uma ferramenta viável para detecção de falhas, mas também flexível e confiável com redução do uso da rede e dos recursos computacionais.

3.2 Arquiteturas orientadas a serviços

SOA é uma técnica emergente, padronizada, que possibilita acoplamento mínimo entre componentes e comunicação distribuída independente de protocolo (PAPAZOGLU; HEUVEL, 2007). O crescente uso desta tecnologia em sistemas de informação, nos últimos anos, foi expandido para o segmento industrial. A facilidade de interconexão e os altos níveis de abstração entre dispositivos fazem com que a tecnologia SOA possa ser empregada também em ambientes industriais com certa facilidade (MORITZ et al., 2008).

Segundo (CANNATA; KARNOUSKOS; TAISCH, 2010), a fábrica do futuro será conduzida pelo alto uso de arquiteturas orientadas a serviços. O gerenciamento dos processos de negócio serão fortemente baseados em informações provenientes do chão de fábrica, tornando as aplicações mais complexas e custosas. Destas necessidades, surgem os conceitos de colaboração entre os diversos segmentos e componentes da aplicação.

Os desafios desta nova abordagem visam prover meios de interconexão entre dispositivos e aplicação. Neste contexto, vários projetos foram iniciados. O projeto SIRENA (Service Infrastructure for Real Time Embedded Networked Applications) foi a primeira iniciativa a aplicar os conceitos de orientação a serviços em dispositivos físicos de chão de fábrica, com foco na implementação distribuída de arquiteturas SOA em ambientes heterogêneos (JAMMES; SMIT, 2005). O projeto investigou os diferentes domínios de aplicação, como automotivo, residencial e telefônico. A partir deste projeto, diversos outros utilizaram os conceitos SOA para integração entre equipamentos em infraestruturas orientadas a serviços (ZEEB et al., 2007).

Um dispositivo que possui recursos para utilização em ambientes SOA apresenta ganhos em interoperabilidade, autonomia além de se tornar uma importante peça na construção da aplicação em que for baseado (CANDIDO et al., 2010b). Dessa forma, a tecnologia empregada não só fornece um padrão de comunicação entre os componentes do sistema, mas também adiciona a capacidade de novas aplicações serem construídas rapidamente,

mantendo a agilidade em relação às modificações impostas por sistemas que apresentam muitas alterações durante o ciclo de vida.

No trabalho de (PATHAK et al., 2007), é apresentada uma arquitetura baseada em serviços para o gerenciamento dos equipamentos de um sistema de transmissão de energia elétrica. A arquitetura possibilita o sensoriamento, integração de informações, avaliação dos riscos e tomada de decisão relativos à operação de um sistema elétrico de alta tensão. Segundo os autores, o sistema proposto integra a aquisição de dados em tempo real, modelagem e funcionalidades de previsão. Juntas, são utilizadas para determinar as políticas de operação, agendar manutenções e garantir o correto funcionamento dos equipamentos que compõem o sistema de distribuição de energia.

O trabalho de (RIBEIRO et al., 2008) mostra o sucesso encontrado em pesquisas anteriores em aplicações que trocam informações na forma de serviços, escondendo a complexidade dos componentes e provendo interfaces limpas de comunicação. No entanto, os autores ilustram cenários onde são inseridas desvantagens com o uso de SOA em sistemas altamente configuráveis. Estes sistemas apresentam a característica de reconfiguração dos dispositivos periodicamente. De acordo com o artigo, a periodicidade da reconfiguração dos dispositivos acontece tipicamente em ambientes heterogêneos, o que acarreta em perda de desempenho do sistema e ocasiona alto tráfego nas redes de comunicação. Como solução para este problema, é apresentado um modelo de comunicação genérica que minimiza os problemas encontrados nestes sistemas.

O artigo de (CANDIDO et al., 2010a) aborda a interoperabilidade entre as especificações OPC Unified Architecture (OPC UA) e DPWS do padrão SOA. Os autores demonstram que, sozinha, nenhuma das duas tecnologias consegue atender a todos os requisitos impostos a nível de dispositivos em aplicações orientadas a serviços, mas, se combinadas, apresentam benefícios para o domínio em que são utilizadas. Uma solução abrangendo as duas tecnologias é proposta e comprova a convergência das duas tecnologias.

Também é visto em (NAGORNY et al., 2013) que o paradigma SOA é uma alternativa promissora na implementação e controle de ambientes ditos sistemas de sistemas. Sistemas de sistemas são coleções de tarefas ou sistemas dedicados que unem seus recursos ou aptidões para desempenhar determinada tarefa a fim de possibilitar a criação de um novo sistema que ofereça mais recursos e possibilidades de desempenhar outras tarefas mais complexas. Como mostra o trabalho, utilizando uma arquitetura orientada a serviços, é possível a integração de sistemas diferentes do ponto de vista estrutural e dinâmico, como, por exemplo, sistema de aquecimento urbano e distribuição de energia. Novamente, o uso e composição dos serviços existentes é feito de forma facilitada, o que contribui para o sucesso da aplicação.

3.3 Uso de arquiteturas orientadas a serviços em conjunto com sistemas de manutenção inteligente

Os trabalhos anteriores apresentam soluções empregadas nas duas áreas distintas. Como apresentado, nos últimos anos, a adoção de sistemas baseados em serviços na indústria está em constante crescimento. Mesmo que a pesquisa nestas duas áreas esteja evoluindo, as iniciativas para o uso de SOA na integração de sistemas de manutenção inteligente ainda são escassas.

O trabalho de (ZHAO et al., 2010a) apresenta o desenvolvimento de uma arquitetura baseada em SOA para monitoramento remoto de condição e diagnóstico de falhas em equipamentos. A arquitetura proposta é construída sobre o conceito de Web Services, clientes inteligentes e XML. O artigo apresenta uma comparação entre programação orientada a objetos, baseada em componentes e SOA, como diferentes técnicas empregadas para construção de aplicações distribuídas. Os autores confirmam o sucesso e vantagens no uso de SOA, verificando a possibilidade de integração de diferentes recursos de software, aumentando a disponibilidade e escalabilidade do sistema e diminuindo o tempo de desenvolvimento.

No artigo de (CANNATA; KARNOUSKOS; TAISCH, 2010), são apresentados os aspectos envolvidos em relação aos conceitos de *e-maintenance* em vistas a cumprir com a construção de dispositivos baseados em SOA. Os autores investigam os benefícios na adoção de arquiteturas orientadas a serviços para a integração de camadas de negócio com sistemas de manutenção inteligente dispostos no chão de fábrica. Através do uso destas tecnologias, é notável a facilidade e benefícios encontrados na integração de sistemas, sugerindo o uso de SOA em uma nova geração de sistemas industriais. Em relação a *e-maintenance*, o monitoramento das condições dos equipamentos de forma facilitada faz com que o gerenciamento de planos de manutenção possam ser executados de maneira ágil e eficiente, evitando paradas desnecessárias na linha de produção.

4 ARQUITETURA PROPOSTA

Conforme visto no capítulo anterior, é possível a integração de um sistema de manutenção inteligente utilizando uma arquitetura orientada a serviços. Dessa forma, este capítulo apresenta a proposta de uma arquitetura para um sistema de manutenção inteligente. São apresentadas as entidades principais e a interligação entre elas através da utilização do padrão SOA. As entidades que compõem a arquitetura são descritas e inseridas no contexto de uma aplicação através de casos de uso. Através dos casos de uso, são abordadas diferentes situações onde as entidades da arquitetura se encaixam para resolver algum dos problemas.

Juntamente com a arquitetura proposta, é apresentada a programação dos experimentos e o estudo de caso. Os experimentos visam validar o sistema proposto, bem como todas as entidades aqui descritas. Para validação, o estudo de caso, envolvendo um conjunto atuador elétrico e válvula, é proposto e hipóteses sobre a utilização adequada do sistema desenvolvido são levantadas.

4.1 Arquitetura orientada a serviços proposta

A arquitetura proposta neste trabalho tem por objetivo a integração de sistemas de manutenção inteligente, dos diversos equipamentos que precisam ser monitorados, além de outras entidades que auxiliam no funcionamento do sistema. A troca de informações entre todos os elementos que compõem a arquitetura é abstraída na forma de serviços, o que facilita a integração, inserção e remoção de novas entidades no sistema. Dessa forma, a especificação da arquitetura é definida utilizando os padrões SOA. Uma visão geral das entidades que fazem parte da arquitetura proposta é apresentada na figura 10. Na figura nota-se que cada entidade tem uma função específica no sistema e o acesso às suas funcionalidades é realizado através de serviços. A seguir, para melhor entendimento, todos os elementos serão detalhados.

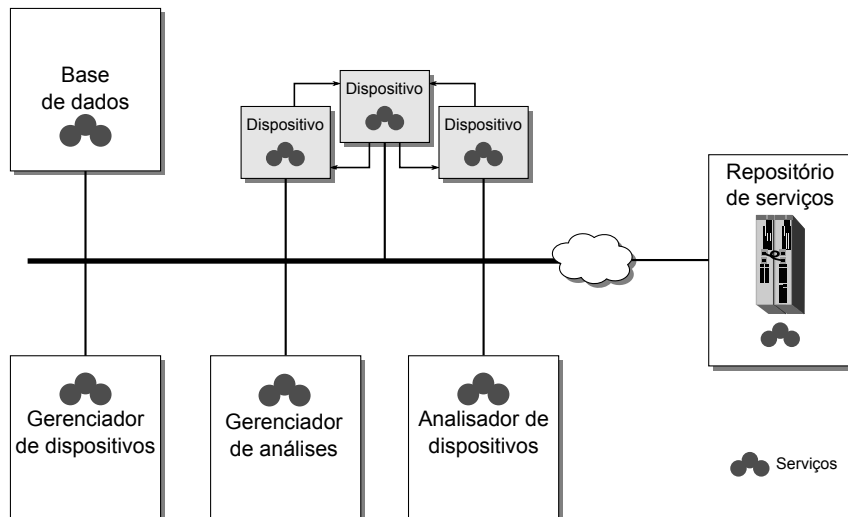


Figura 10: Arquitetura orientada a serviços proposta para integração de um sistema de manutenção inteligente.

4.1.1 Serviço

No contexto da arquitetura proposta neste trabalho, serviço é um componente de software que encapsula uma funcionalidade acessível através dos padrões definidos pela tecnologia SOA. Todos os componentes da arquitetura expõem as suas funcionalidades na forma de serviços, possibilitando que a interação entre eles seja feita de forma transparente. Como parte do padrão SOA, serviços podem ser descobertos e utilizados por clientes que desejam executar uma determinada tarefa. Neste contexto, também é possível a definição de serviços mais especializados com base em outros serviços, praticando a técnica da composição de serviços. Estas características se tornam inerentes à proposta, devido a utilização do padrão SOA.

4.1.2 Dispositivo

Um dispositivo é um componente de software utilizado para encapsular um elemento físico da aplicação proposta. Por ser executado em um dispositivo físico, o componente é denominado dispositivo lógico e disponibiliza serviços para acesso à funcionalidades previamente definidas. As funcionalidades podem ser relativas ao dispositivo físico em que o componente está executando ou outras que auxiliam em alguma tarefa específica não relacionada diretamente com o hardware hospedeiro.

Do ponto de vista da aplicação, os dispositivos são entidades que hospedam serviços. Dentre os serviços hospedados, alguns estão presentes em todos os dispositivos da arquitetura, servindo de base para a comunicação entre todos os elementos desta classe. Dessa forma, conhecendo os serviços básicos, uma interface mínima para troca de informações entre as entidades do sistema é definida, facilitando a inserção de novos dispositivos.

Além dos serviços base, outros podem ser executados no dispositivo. A arquitetura

permite o envio de novos serviços para os dispositivos do sistema. O dispositivo recebe o novo serviço, que possui as mesmas funcionalidades de um serviço padrão, e o carrega para ser executado normalmente.

4.1.3 Gerenciador de Dispositivos

Como o nome sugere, o Gerenciador de Dispositivos é um componente de software utilizado para gerenciar os dispositivos da arquitetura proposta. A configuração ou obtenção da lista dos dispositivos na aplicação orientada a serviços são exemplos de funcionalidades do gerenciador. Como os dispositivos são entidades que mapeiam funcionalidades dos dispositivos físicos encontrados em um sistema de manutenção inteligente, o gerenciador pode, por exemplo, obter a lista de sensores do equipamento e configurar alguns parâmetros, como a taxa de atualização dos dados dos sensores. Também é possível definir comportamentos adicionais para o caso do equipamento estar operando em diferentes níveis de degradação. Os diferentes comportamentos podem ser utilizados quando há necessidade de operar o equipamento em condições onde manter os níveis de degradação estáveis é mais importante do que a operação a pleno. Portanto, pode-se manter o equipamento funcionando até que uma manutenção *in loco* possa ser realizada.

Grupos de dispositivos podem ser criados e gerenciados. O Gerenciador de Dispositivos permite a criação de grupos, a fim de facilitar a alteração simultânea de vários dispositivos. Supondo que a aplicação é inerentemente escalável, o número de dispositivos tende a aumentar, o que dificulta o gerenciamento ou configuração de dispositivos similares. Considerando que as configurações aplicadas a uma mesma classe de dispositivos é muito parecida, o gerenciador permite que um grupo receba os mesmos parâmetros, automatizando o processo de customização dos dispositivos.

Outra funcionalidade que o gerenciador provê é a obtenção da topologia dos componentes do sistema. Em uma arquitetura SOA todos os serviços estão localizados no mesmo nível hierárquico. Essa característica pode ser considerada positiva do ponto de vista de integração e reuso de serviços. Em contrapartida, ao definir níveis hierárquicos entre componentes de um mesmo dispositivo físico, a complexidade para o estabelecimento de uma hierarquia lógica para representação das diferentes partes desse dispositivo aumenta. Contudo, ao aplicar um identificador único para cada dispositivo, ou parte de dispositivo, do sistema, torna-se possível definir e estabelecer uma relação hierárquica. O identificador pode ser uma URI, por exemplo, onde cada parte do endereço se refere a um dos níveis hierárquicos.

O Gerenciador de Dispositivos também é utilizado na resolução de problemas encontrados durante a execução dos componentes do sistema. A entidade se enquadra na categoria de Interface Homem-Máquina (IHM), possibilitando que a rede seja mapeada em busca de dispositivos ou serviços de forma interativa. Além disso, é possível obter o

estado dos dispositivo, visualizar dados ou testar os serviços encontrados.

4.1.4 Gerenciador de Análises

O Gerenciador de Análises é utilizado para definir o plano que será aplicado a determinado equipamento, ou grupo de equipamentos, tendo em vista obter os níveis de degradação. No contexto deste trabalho, um plano de análise é uma estrutura que armazena as informações necessárias para que uma análise possa ser executada corretamente. Cada plano armazena as ferramentas que serão utilizadas para análise de dados, bem como a ordem em que serão executadas. O gerenciador permite o agendamento dos planos e a execução periódica em intervalos de tempo. Os planos também podem ser gerenciados, possibilitando alteração e/ou exclusão.

No plano também são definidos os comportamentos a serem adotados com base nos valores de degradação do equipamento. O gerenciador obtém os comportamentos previamente definidos para o equipamento a ser analisado, os quais podem ser mapeados pelo usuário para diferentes níveis de degradação. Dessa forma, o equipamento pode ser adaptado a diferentes condições, por exemplo, evitando o aumento da degradação até que uma manutenção possa ser realizada.

No geral, informações como nome, descrição e dispositivos afetados fazem parte do plano de análise. Estas informações permitem ao operador do sistema identificar os planos criados e onde estão empregados. Como o plano armazena os dispositivos que serão analisados, também é possível identificar a origem dos dados utilizados. Dessa forma, pode-se mapear os sensores dos equipamentos como fonte de dados para alimentar as ferramentas de análise. Portanto, a origem dos dados utilizados também está contida no plano. Outra informação que faz parte desta estrutura é referente às ferramentas utilizadas na análise. Além delas, o plano armazena a ordem em que serão utilizadas, a qual é definida pelo operador do sistema.

O Gerenciador de Análises também coordena as análises realizadas em um grupo de equipamentos similares. Em alguns casos, o processo de obtenção dos valores de confiança de um conjunto de equipamentos pode ser simplificado pelo fato de que, se são similares e estão sujeitos às mesmas condições de degradação, o estado de saúde de um equipamento pode ser aproximado para o estado do grupo. Dessa maneira, é possível elaborar um plano de análise que seja aplicável a vários equipamentos.

4.1.5 Analisador de Dispositivos

O Analisador de Dispositivos concentra todas as ferramentas necessárias para a realização da análise dos dados de determinado dispositivo. A entidade opera com base nos planos criados e agendados pelo operador do sistema no Gerenciador de Análises. De posse do plano, o analisador executa as análises utilizando os algoritmos necessários

para obtenção do resultado esperado. A execução de um mesmo plano é feita de forma automática.

O analisador permite verificar quais planos estão sendo executados ou que aguardam na fila para execução. Sempre que um novo plano é gerado, o analisador o coloca na fila de execução, a fim de que seja finalizado o mais breve possível. A entidade permite o monitoramento destes planos pelo usuário, o qual pode verificar se a execução está de acordo com o esperado.

4.1.6 Base de dados

A entidade Base de Dados é utilizada como uma interface para um banco de dados. O software encapsula funcionalidades para armazenar todos os dados relativos à aplicação. Desde os dados obtidos nas análises de degradação dos equipamentos como também os planos de análise agendados pelo usuário. A base centraliza as informações, mas não necessariamente faz com que a aplicação seja centralizada. Nada impede a utilização de diferentes bases de dados e o compartilhamento de informações entre as entidade do sistema. Por exemplo, o armazenamento dos dados de treinamento de um equipamento, necessário para obtenção do nível de degradação, pode ser feito em diferentes bases de dados. O Analisador de Dispositivos pode combinar estas informações, a fim de melhorar o resultado esperado.

A Base de Dados é definida como um componente SOA, portanto o acesso aos dados é feito através de serviços. A inclusão, deleção e modificação de dados são realizadas por serviços especializados. Um dispositivo remoto pode acessar a base de dados a fim de obter o histórico dos últimos valores de confiança calculados ou valores de testes intermediários para determinado equipamento, utilizando-os para um novo tipo de análise.

4.1.7 Repositório de serviços

O repositório de serviços é uma entidade que armazena os dispositivos lógicos e seus serviços, os quais podem ser obtidos dinamicamente e implantados em dispositivos físicos. Esta entidade pode estar localizada na rede local ou remota, com acesso disponível para várias aplicações. Dessa forma, uma nova aplicação, isolada de outra já existente, pode ser construída com o reuso de componentes obtidos de um repositório de serviços compartilhado pelas duas. Esta abordagem também permite o compartilhamento de uma base de dados de componentes lógicos entre aplicações.

4.2 Casos de uso para a arquitetura proposta

Os elementos que compõem a arquitetura proposta encaixam-se em alguns casos de uso no contexto de um sistema de manutenção inteligente. Os casos de uso propostos

a seguir exemplificam a interação do operador do sistema na busca ou configuração dos dispositivos e na criação dos planos de análise para cada dispositivo. Para a maioria dos casos de uso, o operador do sistema é considerado como ator. Além disso a ferramenta de análise de dados é utilizada para analisar os dados obtidos dos equipamentos monitorados. Visto que as análises executam autonomamente durante o processo de amostragem dos equipamentos, uma tarefa periódica é vista como ator do caso de uso.

4.2.1 Descoberta e configuração de dispositivos

A descoberta dos dispositivos e serviços na rede pode ser feita a qualquer momento utilizando o Gerenciador de Dispositivos. Cada um dos dispositivos encontrados é identificado e colocado em uma lista de dispositivos ativos. Também são identificados os serviços hospedados em cada um dos dispositivos. Dessa forma, é possível determinar quais dispositivos estão atualmente disponíveis na rede e quais possuem serviços que poderão ser utilizados para suprir e executar determinada funcionalidade do sistema.

A geração da hierarquia de recursos encontrados na rede também é possível. Ao aplicar um identificador único para cada recurso, a reconstrução dos componentes do sistema pode ser obtida agrupando-os em classes ou categorias. O identificador é utilizado para auxiliar a definir uma topologia dos subdispositivos encontrados em um dispositivo lógico, facilitando a visualização e configuração dos componentes.

O diagrama Unified Modeling Language (UML) de casos de uso para descoberta e configuração de dispositivos é apresentado na figura 11. São ilustradas as interações que o operador do sistema pode realizar nos dispositivos utilizando o Gerenciador de Dispositivos. Em via de regra, o gerenciador será utilizado pelo operador do sistema para descobrir os dispositivos da rede, obter o estado ou realizar modificações nestes elementos.

Ao utilizar a descoberta de dispositivos, o gerenciador analisa todos dispositivos da aplicação orientada a serviços. Os que forem encontrados são retornados, permitindo ao operador do sistema verificar se a inclusão de um novo dispositivo foi concluída com êxito. Além disso, o gerenciador permite obter o estado de cada elemento encontrado, o qual está relacionado ao funcionamento do dispositivo, se está pronto para operação ou executando corretamente. Outras informações também são obtidas utilizando os metadados. Esta funcionalidade lista dados como modelo, versão ou fabricante do dispositivo.

Na configuração, o operador do sistema também deve informar os serviços de monitoramento dos sensores do equipamento para enviar os dados à base de dados. Com as informações de acesso à base de dados, o dispositivo pode enviar automaticamente os dados adquiridos dos sensores. Para evitar o excesso de acessos à base, um montante de dados é adquirido e armazenado localmente pelo dispositivo. Quando o montante estiver completo, é enviado diretamente para a base de dados. É possibilitado ao operador do

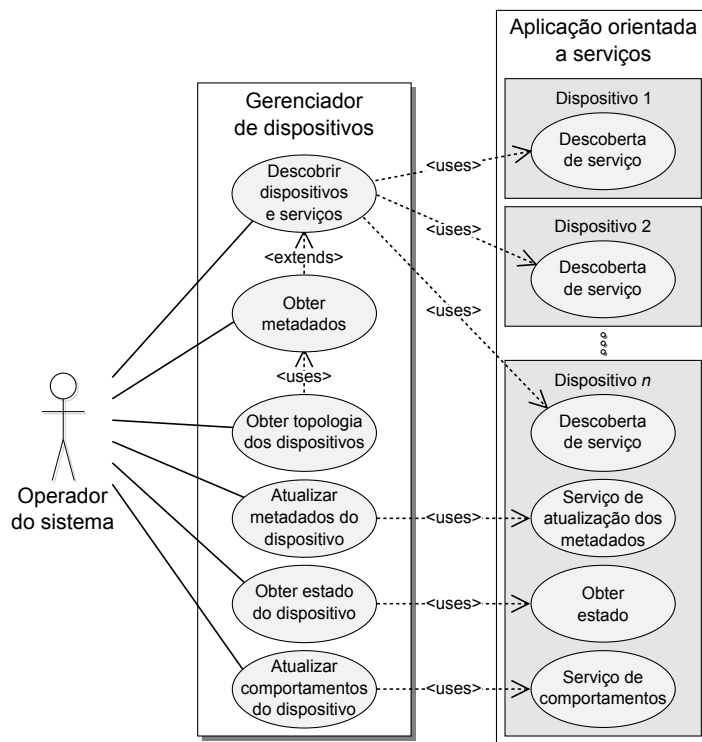


Figura 11: Diagrama de casos de uso para descoberta e configuração de dispositivos.

sistema configurar o número de amostras que o dispositivo irá adquirir para compor o conjunto de dados. A figura 12 apresenta o diagrama UML de casos de uso que ilustra a configuração do dispositivo para envio dos dados para a Base de Dados.

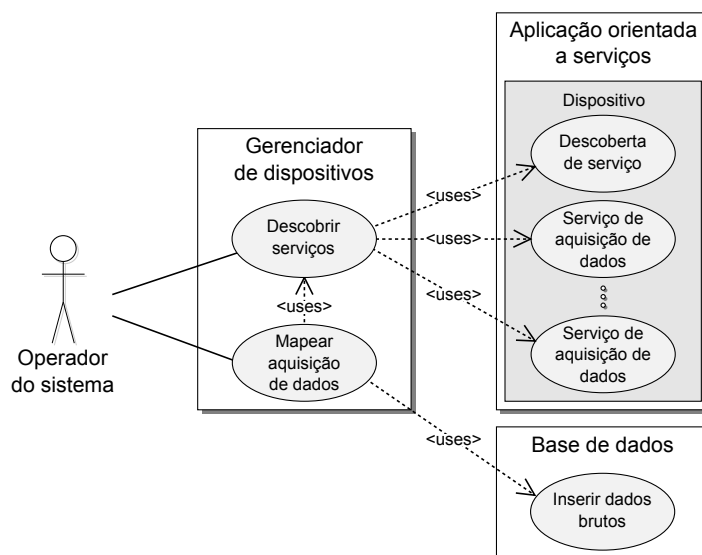


Figura 12: Diagrama de casos de uso para a configuração do envio dos dados do dispositivo.

Outra funcionalidade que o Gerenciador de Dispositivos disponibiliza é a obtenção da topologia da distribuição dos dispositivos na rede. Dispositivos e serviços no padrão SOA sempre estão localizados no mesmo nível hierárquico, o que dificulta o estabelecimento

de uma hierarquia entre os componentes. No entanto, pode-se definir uma hierarquia utilizando para isso o identificador de cada dispositivo. O identificador – uma URI, por exemplo – é único para cada dispositivo da aplicação, possibilitando que, na inclusão de subdispositivos, o identificador seja estendido para englobá-los. Além disso, pode-se criar classes e subclasses de dispositivos, organizando funcionalidades comuns para determinado dispositivo. A visualização das duas topologias possíveis é apresentada na figura 13. Em (a), os dispositivos estão distribuídos como são encontrados pelo Gerenciador de Dispositivos. Esta visão pode ser considerada a visão do ponto de vista da rede. Em (b), o gerenciador reconstrói a topologia baseado nos identificadores únicos de cada dispositivo. Esta visão pode ser considerada a visão lógica do sistema. Nota-se que os dispositivos, que na primeira topologia estão no mesmo nível hierárquico, agora são encontrados como subdispositivos de outros dispositivos.

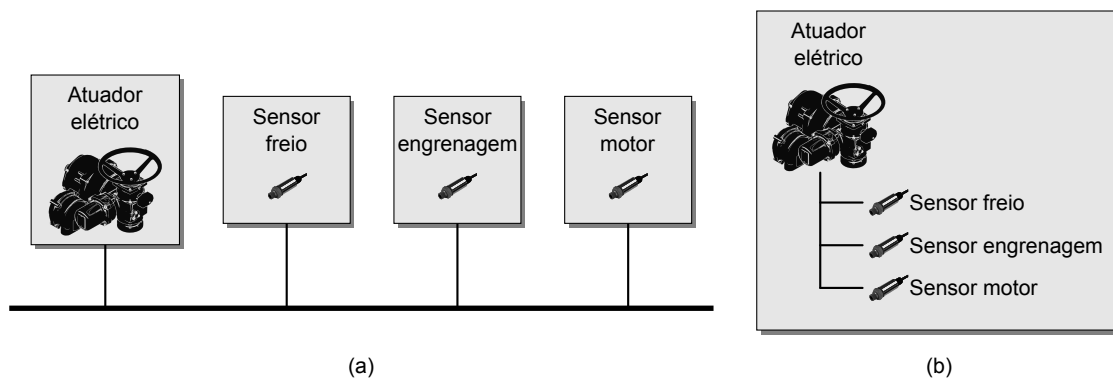


Figura 13: Possíveis visualizações da topologia dos dispositivos: (a) visão da rede; (b) visão lógica.

Tomando como exemplo a figura 13, os identificadores de cada dispositivo poderiam ser construídos com um dispositivo pai englobando outros dois subdispositivos. Tanto o dispositivo pai como os subdispositivos estão definidos dentro de escopos ou áreas e essas informações estão incluídas no identificador. Por exemplo, os dispositivos sensores (*Sensor freio*, *Sensor engrenagem*, *Sensor motor*) logicamente fazem parte do dispositivo *Atuador elétrico*. Dessa forma, é possível a construção de um identificador para cada um deles, como segue:

- Atuator elétrico:

`http://system.com/equipment/ElectricActuator`

- Sensor freio:

`http://system.com/equipment/ElectricActuator/BrakeSensor`

- Sensor engrenagem:

`http://system.com/equipment/ElectricActuator/GearSensor`

- Sensor motor:

`http://system.com/equipment/ElectricActuator/MotorSensor`

Com o exemplo, fica claro como os componentes são distribuídos hierarquicamente no sistema. Através do endereço lógico, os sensores são agrupados em um nível hierárquico abaixo do dispositivo *Atuador elétrico*. Ao incluir mais dispositivos neste nível, o padrão de organização do identificador deve seguir o mesmo formato. Também é possível que dispositivos abaixo do nível hierárquico dos sensores sejam adicionados. Dessa forma, ao final do endereço do sensor deve-se adicionar um separador e o novo identificador.

Outra forma de organizar hierarquicamente os dispositivos é por meio de grupos. Ainda referente ao exemplo do atuador elétrico, caso fosse composto por sensores e atuadores, novos separadores de grupos, definidos como *sensors* e *actuators*, poderiam ser adicionados ao endereço das entidades. Dessa forma, os dispositivos distribuídos logicamente no nível hierárquico mais baixo estariam organizados da seguinte maneira:

- Sensores do atuador elétrico:

`http://system.com/equipment/ElectricActuator/sensors`

- Atuadores do atuador elétrico:

`http://system.com/equipment/ElectricActuator/actuators`

Quanto à configuração, cada dispositivo pode armazenar diferentes comportamentos, os quais dizem respeito ao modo de operação que o equipamento vai assumir perante determinada situação. Os comportamentos são definidos pelo operador do sistema. São diferentes tarefas que podem ser executadas pelo equipamento em situações específicas. No caso de verificação de degradação excessiva de um equipamento, é possível alterar o comportamento de funcionamento para outro que priorize a manutenção do estado de saúde atual, evitando o aumento da degradação. Dessa forma, o equipamento pode operar por um período maior de tempo até que uma manutenção possa ser realizada. Contudo, medidas que alteram o comportamento podem levar à perda de desempenho nas tarefas executadas. Portanto, o operador do sistema deve ponderar a escolha e utilização adequada de cada um, visando manter a qualidade do nível de operação desejado.

4.2.2 Envio de dados de treinamento

Para obter o índice de degradação dos equipamentos, as ferramentas de análise necessitam comparar os dados obtidos durante o processo com dados de treinamento. Dentre os dados de treinamento, emergem duas categorias: normal e falha. Os dados de funcionamento normal são obtidos quando o equipamento está funcionando normalmente, enquanto que os dados de falha são obtidos quando o equipamento está funcionando com algum tipo de degradação. A aquisição de ambos os tipos de dados são feitas em lugar apropriado, tendo certeza das características de cada sinal.

O envio de dados de treinamento é feito através do Gerenciador de Dispositivos. O gerenciador permite que o operador do sistema envie os dois tipos de dados, mapeando-os

para um dispositivo ou para um grupo de dispositivos. A figura 14 apresenta os casos de uso para o envio de dados de treinamento a um dispositivo.

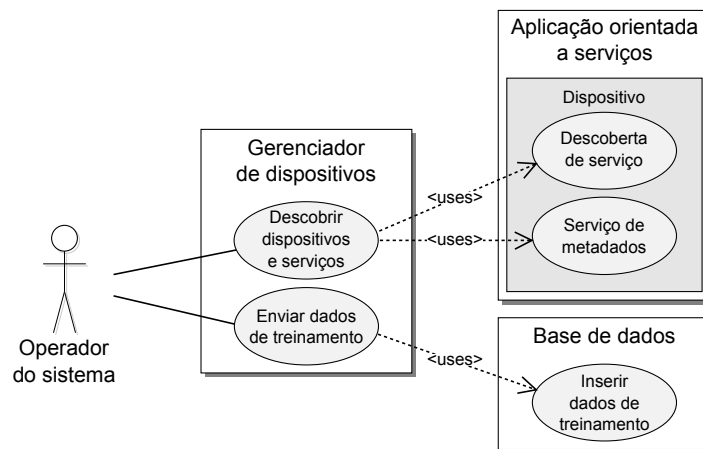


Figura 14: Diagrama de casos de uso para envio de dados de treinamento para um dispositivo.

4.2.3 Gerenciamento de análises

O gerenciamento de análises é feito pelo operador do sistema utilizando a entidade Gerenciador de Análises. As análises são definidas em planos, os quais apresentam informações sobre o dispositivo analisado, quais dados serão utilizados e as ferramentas empregadas na manipulação dos dados. É permitido o gerenciamento dos planos, tendo em vista a edição ou remoção, por parte do operador do sistema. Como os planos são armazenadas em uma base de dados, facilmente podem ser compartilhados com outras instâncias do Gerenciador de Análises e utilizados por outros operadores do sistema.

As informações contidas em um plano de análise ditam como determinado equipamento será analisado. Em um primeiro momento, o operador do sistema faz a busca pelos dispositivos utilizando o Gerenciador de Dispositivos. Com a lista de dispositivos ativos, o Gerenciador de Análises é utilizado para definir as operações a serem realizadas em um equipamento específico. Preenchendo o plano de análise, o operador seleciona quais dados do equipamento serão utilizados e quais as ferramentas utilizadas para analisá-los. O plano também permite definir o intervalo de execução de cada análise e por quanto tempo o equipamento será analisado. O diagrama UML de casos de uso para algumas funcionalidades do Gerenciador de Análises é apresentado na figura 15. A estrutura do plano é armazenada na entidade Base de Dados, sendo acessível para as outras entidades que utilizam as informações do plano.

Na construção do plano de análise, os comportamentos de cada dispositivo são obtidos e incluídos na execução. Os diferentes comportamentos podem ser mapeados para intervalos de resultados esperados das análises. Como descrito pelo diagrama de casos de uso da figura 15, a figura 16 ilustra a criação de um plano de análise de dispositivo.

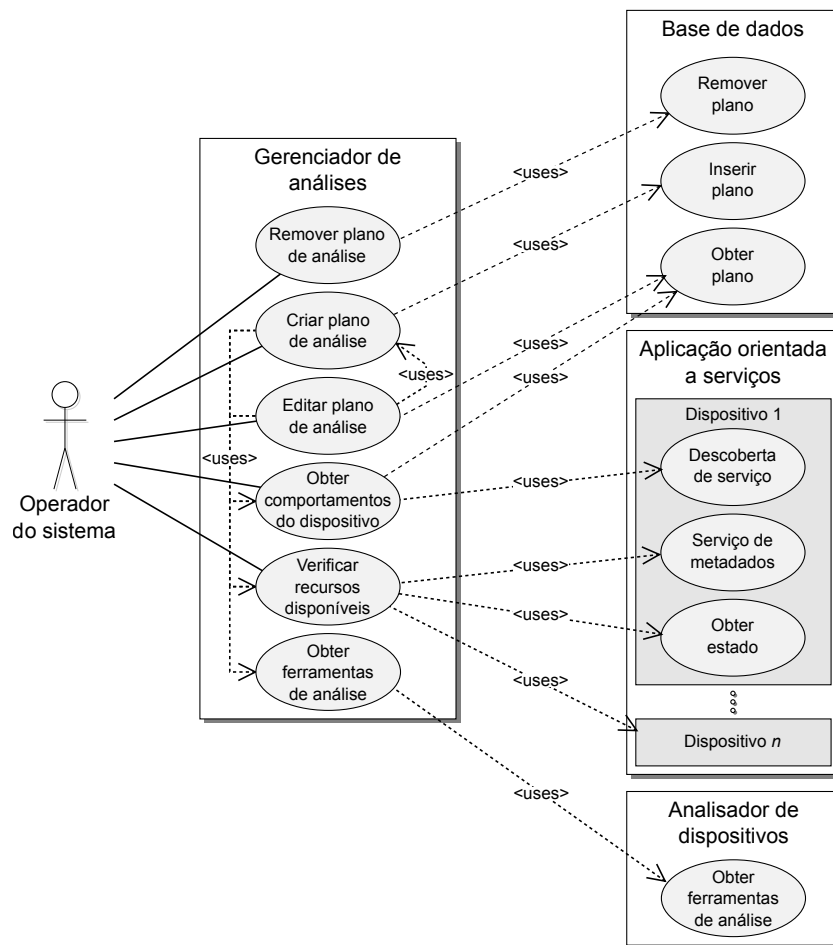


Figura 15: Diagrama de casos de uso para o gerenciamento de análises pelo operador do sistema.

Primeiramente, o operador do sistema deve utilizar o Gerenciador de Análises (1). Após a execução do gerenciador, é possível realizar a busca pelos dispositivos na rede (2). Os dispositivos encontrados são listados para o operador, bem como outros subdispositivos que o compõe. No exemplo, fazem parte da rede os dispositivos *Atuador elétrico*, *Sensor freio*, *Sensor engrenagem* e *Sensor motor*. Contudo, os sensores, que na topologia de rede estão no mesmo nível hierárquico, fazem parte do dispositivo *Atuador elétrico*.

Ao obter os dispositivos da rede, o operador do sistema pode criar um plano de análise (3). O plano é criado também utilizando o Gerenciador de Análises, como ilustrado ainda na figura 16. No plano, é definido qual dispositivo será analisado, quais sensores serão monitorados e as ferramentas utilizadas na análise dos dados. Após a criação do plano, o operador tem a possibilidade de armazená-lo (4). Sendo assim, o plano é enviado para a base de dados, onde estará visível para utilização pelo Analisador de Dispositivos ou mesmo para visualização ou modificação posterior por parte do operador.

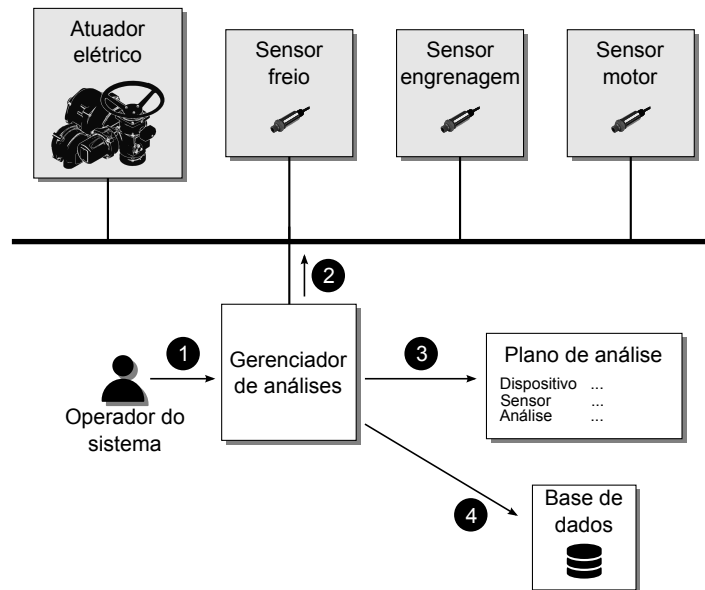


Figura 16: Busca de dispositivo e criação de um plano de análise.

4.2.4 Análise dos dados do dispositivo

As análises dos equipamentos do sistema, anteriormente agendadas na ferramentas de Gerenciamento de Análises, são executadas pela entidade Analisador de Dispositivos. O analisador obtém os planos e os executa conforme programado pelo operador do sistema. Conhecendo a frequência em que cada plano deve ser executado, uma tarefa periódica é definida. Dessa forma, chegada a hora da reexecução de um plano, a tarefa se comunica com o Gerenciador de Análises e o dispara. Os possíveis casos de uso para a tarefa periódica são apresentados no diagrama UML da figura 17. A tarefa tem a possibilidade de obter os planos de análise ativos ou selecionar diretamente um plano conhecido. Após a carga, a tarefa pode iniciar a execução do plano. Na execução, o Analisador de Dispositivos seleciona as ferramentas de análise, busca os dados do equipamento e, ao final, armazena os resultados na Base de Dados.

A figura 18 apresenta o procedimento utilizado quando da execução de um plano de análise pelo Analisador de Dispositivos. O analisador verifica a base de dados em busca dos planos ativos (1). Ao encontrá-los, o analisador os analisa verificando para qual dispositivo o plano deve ser empregado (2). Com o plano, o Analisador de Dispositivos busca novamente os dados dos sensores do dispositivo, indicados no plano e que estão armazenados na base de dados (3). Ao término da busca, os dados são analisados e o resultado da análise é novamente enviado para a base de dados (4). Se necessário, novas análises são realizadas sobre os dados que já foram manipulados. Com o índice de degradação, que é o resultado final da análise, o Analisador de Dispositivos seleciona um dos comportamentos definidos no plano. O equipamento é então posto para executar a tarefa monitorada de acordo com o novo comportamento (5).

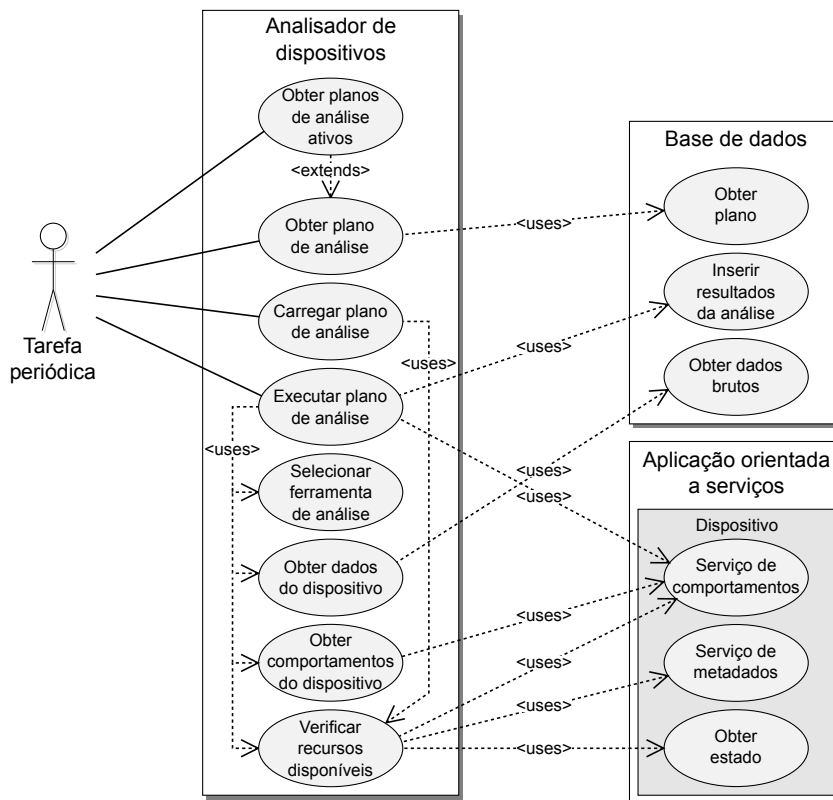


Figura 17: Diagrama de casos de uso para a obtenção e execução das análises.

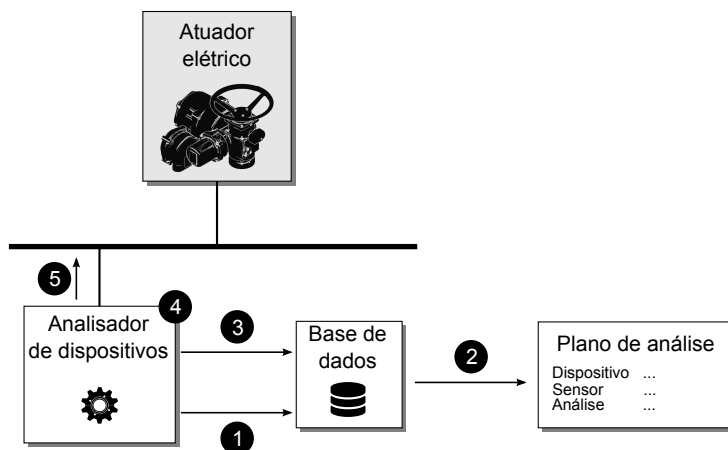


Figura 18: Diagrama de execução de uma análise de equipamento.

4.2.5 Relatórios de saúde

A arquitetura proposta permite a obtenção dos relatórios de saúde dos dispositivos executando sobre equipamentos monitorados pelo sistema de manutenção inteligente. Como todos os dados das análises são armazenados na entidade Base de Dados, é possível, a qualquer momento, resgatá-los. A obtenção dos relatórios dos dispositivos é possível com o uso do Gerenciador de Dispositivos. Os casos de uso para obtenção dos relatórios de um dispositivo ou de um grupo de dispositivos são apresentados no diagrama UML da figura 19.

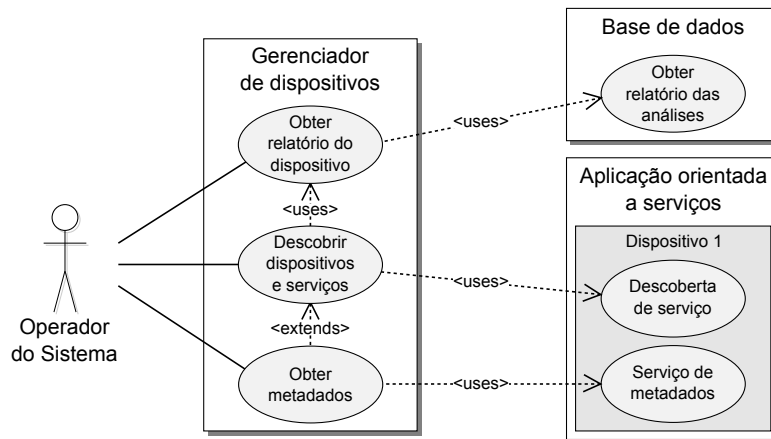


Figura 19: Diagrama de casos de uso para obtenção dos relatórios de saúde do dispositivo.

4.3 Estudo de caso

O estudo de caso utilizado neste trabalho consiste de um conjunto atuador elétrico e válvula. O atuador é utilizado para controle de fluxo em redes de distribuição de petróleo. Os atuadores permitem a abertura e fechamento dos dutos através do movimento da haste pelo acionamento de um motor. Devido a natureza do equipamento, é inerente a predisposição à degradação.

Como parte do trabalho, o atuador é definido como um dispositivo da arquitetura proposta. São mapeadas as informações pertinentes e traduzidas para o projeto de um cliente. O cliente será um software para simulação de dados, não sendo embarcado no equipamento físico.

Para o projeto do software de simulação do atuador, foram utilizados dados de operação do equipamento real. Os dados foram obtidos através de instrumentação feita nos trabalhos anteriores de (BÖESCH, 2011) e (FACCIN, 2011). Com os dados reais, o software deve utilizá-los para simular a aquisição de dados do equipamento em funcionamento.

4.3.1 Visão geral do objeto de estudo de caso

O objeto de estudo de caso deste trabalho é um conjunto atuador elétrico e válvula. Um atuador elétrico permite a abertura e fechamento motorizado de válvulas através da movimentação de uma haste. Dessa forma, válvulas são encontradas em diversas aplicações onde seja necessário o controle de fluxo de fluidos, como, por exemplo, água, esgoto ou petróleo. Para cada aplicação, é indicado um tipo diferente de válvula. Os tipos mais comuns são esfera, gaveta e globo (CAMPOS; TEIXEIRA, 2006).

O conjunto atuador elétrico e válvula, utilizado no trabalho em questão, é utilizado para controle de fluxo em redes de distribuição de petróleo, modelo CSR6, fabricado pela empresa Coester Automação Ltda. A figura 20 ilustra o atuador utilizado. É possível a obtenção do torque mecânico exercido pelo atuador sobre as engrenagens de comando

da haste bem como a posição do obturador. O acionamento do conjunto pode ser feito localmente, através de uma interface de programação local, sendo que todos os elementos de controle estão incorporados no próprio equipamento. O atuador utilizado é dito de comando inteligente, pois, além de estar instrumentado, possui a capacidade de detecção de alguns problemas, como sobreaquecimento, torque excessivo ou falta de fase.



Figura 20: Atuador elétrico modelo CSR6.

Durante o funcionamento, os valores de torque e posição da haste são armazenados em uma memória interna do atuador, sendo possível registrar até 500 operações de fechamento ou abertura. No entanto, as medidas obtidas pelo atuador não são suficientes para utilização em um sistema de detecção de falhas. Uma das causas é a baixa resolução com que os dados são adquiridos, sendo de 1 Nm e a cada 5% de abertura para um fundo de escala de 60 Nm. Além disso, como forma de detectar facilmente o início e o fim do movimento do atuador, o torque é truncado para 10 Nm quando os valores forem menores que 10 Nm e, quando não há movimento, assume-se o valor de 0 Nm (LAZZARETTI, 2012).

O atuador foi montado em uma bancada de testes onde é possível a simulação das condições de desgaste do conjunto e a instrumentação de partes que não são verificadas por padrão (BÖESCH, 2011; FACCIN, 2011; LAZZARETTI, 2012). Dessa forma, na bancada, o atuador está instrumentado com mais três sensores, visando a obtenção dos níveis de vibração da estrutura. Os sensores são acelerômetros, posicionados no eixo do motor do atuador, na ponta do sem-fim e na pinça do freio. A figura 21 ilustra o posicionamento dos sensores instalados e a estrutura interna do atuador elétrico.

A bancada também conta com um sistema de freio a disco instalado no eixo do atuador, o que possibilita a simulação dos esforços mecânicos exercidos pelo fluxo de fluido durante a abertura e fechamento da válvula. A simulação do desgaste é feita pela troca de engrenagens com diferentes níveis de degradação. No total são três engrenagens, uma em estado normal e outras duas apresentando falha. Das engrenagens com falha, uma está

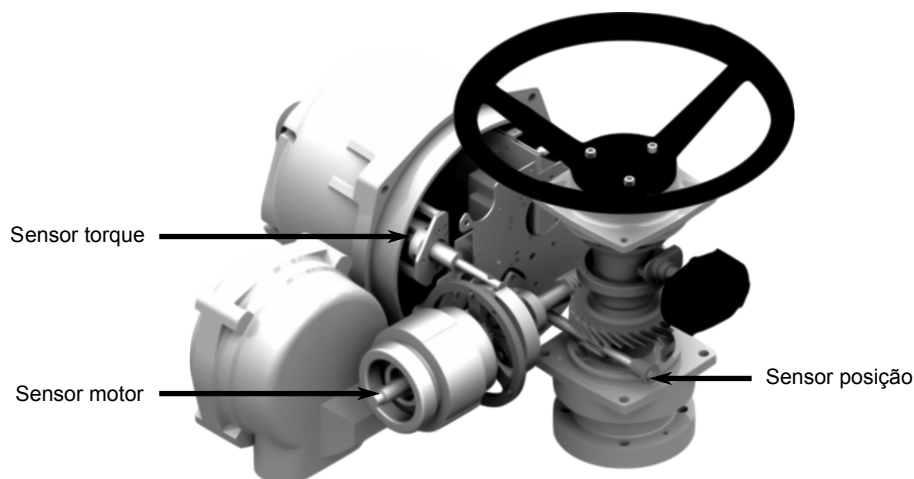


Figura 21: Diagrama da localização dos sensores instalados no atuador elétrico.

desgastada e a outra possui uma parte quebrada. As três engrenagens utilizadas no estudo de caso são ilustradas na figura 22.



Figura 22: Engrenagens utilizadas no estudo de caso – adaptado de (LAZZARETTI, 2012).

As engrenagens fazem parte do conjunto de engrenagens satélite do atuador. Este conjunto é responsável pela transmissão do movimento do motor para a haste da válvula. A posição do conjunto de engrenagens satélite é apresentada na figura 23. O sensor de torque, que faz parte do conjunto sem necessidade de modificação na estrutura, está acoplado ao eixo de acionamento do torque.

4.3.2 Definição dos dispositivos

No contexto deste trabalho, um dispositivo é a entidade lógica principal que abstrai um elemento da aplicação. Pode representar uma entidade física, como um sensor ou atuador, ou lógica, como uma máquina, composta por diversas entidades de hardware mapeadas como entidades lógicas. Cada um dos dispositivos hospeda serviços, representando tarefas ou funcionalidades específicas possíveis de serem executadas. Tanto os dispositivos, bem como os serviços por eles hospedados, podem ser descobertos e identificados na rede. Os dispositivos podem descobrir outros dispositivos e utilizar os serviços do

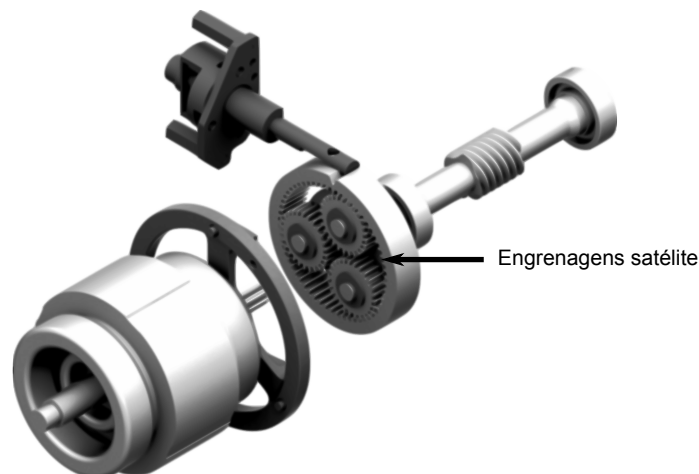


Figura 23: Localização das engrenagens satélite no atuador elétrico.

segundo, a fim de criar um serviço composto mais complexo para execução de determinada tarefa. A figura 24 ilustra a topologia para os dispositivos utilizada neste trabalho. Os clientes utilizam os serviços hospedados pelos dispositivos diretamente. Um serviço também pode ser considerado um cliente caso utilize de uma funcionalidade remota para prover a sua funcionalidade ou tarefa. Além disso, um dispositivo também pode ser considerado um cliente, o que flexibiliza a integração entre os componentes do sistema.

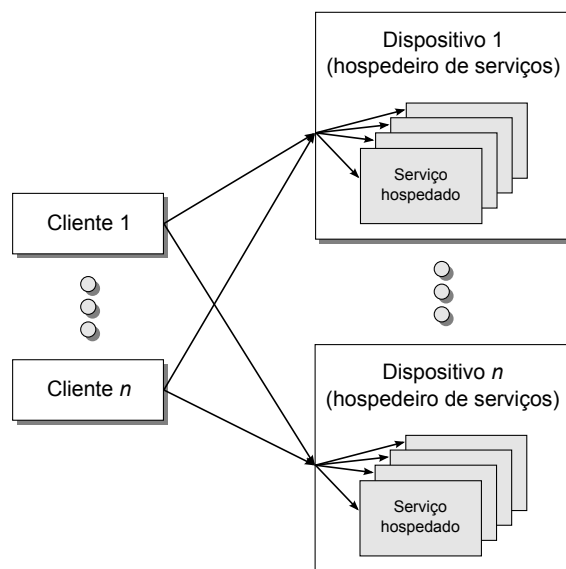


Figura 24: Visão geral dos clientes, dispositivos e serviços hospedados.

O conjunto utilizado como estudo de caso foi mapeado para a representação de dispositivos da arquitetura proposta. Foram verificadas algumas propriedades passíveis de serem incluídas em todos os dispositivos que integram o sistema. Todas as informações mapeadas são acessíveis em forma de serviço. Dessa forma, permitindo a integração do dispositivo com as diversas entidades de software projetadas. A tabela 1 apresenta as informações relativas aos dispositivos da arquitetura proposta. Têm-se informações sobre o

equipamento, como nome, modelo e número de série, e também sobre o seu fabricante.

Tabela 1: Metadados disponíveis para configuração em cada dispositivo.

| Parâmetros de metadados | Tipo |
|-------------------------|--------|
| Fabricante | String |
| Fabricante (URL) | URI |
| Modelo (nome) | String |
| Modelo (número) | String |
| Modelo (URL) | URI |
| Nome do dispositivo | String |
| Número serial | String |
| URL de apresentação | URI |

Da mesma forma que para os dispositivos, os sensores que compõem o equipamento também possuem informações que devem estar disponíveis para a aplicação. Os sensores representam componentes importantes no sistema proposto, pois é onde os dados utilizados nas análises são gerados. Como exemplo, são utilizados pelo Gerenciador de Análises para definir um plano de análise. As informações dos sensores, que estão presentes em todos os elementos deste tipo na arquitetura proposta, são apresentadas na tabela 2. Dentre as informações, estão o modelo, versão e descrição, além do fabricante e a unidade de medida disponibilizada para os dados adquiridos.

Tabela 2: Parâmetros configuráveis dos sensores.

| Parâmetro | Tipo |
|-------------------|--------|
| Fabricante | String |
| Modelo | String |
| Versão | String |
| Descrição | String |
| Unidade de medida | String |

4.3.3 Aquisição dos dados

Os dados foram obtidos de acordo com os trabalhos de (BÖESCH, 2011) e (FACCIN, 2011). Segundo os autores, a aquisição dos dados foi feita por meio de um sistema desenvolvido em LabVIEW utilizando o chassis cRIO-9104 e o módulo para aquisição dos dados de vibração NI-9233. Os dados foram coletados durante seis situações distintas, onde, para cada situação, foram obtidos 25 conjuntos de dados de abertura e 25 de fechamento da válvula, o que representa um total de 300 curvas de experimentos. As situações foram: utilização de engrenagens em perfeito estado e sem acionamento do freio mecânico; engrenagens em perfeito estado e acionamento do freio mecânico com pressão de 1 bar; engrenagens em perfeito estado e acionamento do freio mecânico com pressão de

3 bar; utilização de uma engrenagem desgastada sem acionamento do freio; três engrenagens desgastadas sem acionamento do freio; uma das engrenagens com dentes quebrados sem acionamento do freio mecânico. A tabela 3 ilustra as seis situações de coleta dos dados.

Tabela 3: Situações definidas para coleta dos dados.

| Situação | Acionamento do freio | Engrenagens |
|----------|----------------------|---------------|
| 1 | – | – |
| 2 | 1 bar | – |
| 3 | 3 bar | – |
| 4 | – | 1 desgastada |
| 5 | – | 3 desgastadas |
| 6 | – | 1 quebrada |

Os dados de cada experimento foram amostrados a uma taxa de 2048 S/s. Dessa forma, cada conjunto de abertura ou fechamento da válvula tem duração aproximada de 45 s. Para tornar o processamento dos dados menos custoso, uma subamostragem para a taxa de 1024 S/s foi realizada. Segundo (FACCIN, 2011), a subamostragem dos dados não prejudica o resultado da análise, porém torna o processamento menos custoso.

4.3.4 Análise de degradação

A análise de degradação do atuador é feita utilizando somente um dos três sensores citados anteriormente. De acordo com os trabalhos de (BÖESCH, 2011) e (FACCIN, 2011), o acelerômetro acoplado ao eixo do motor é o que apresenta melhores resultados para a detecção na degradação das engrenagens do atuador. Como apresentado em (LAZZARETTI, 2012), os dados que apresentam melhores resultados para a obtenção dos níveis de degradação são as curvas de abertura da válvula, devido à maior força que é exercida na sede da válvula.

Os sinais de vibração não são estacionários. Para analisá-los, obtendo as características variantes no tempo, técnicas de processamento de tempo e frequência são as mais indicadas. Portanto, a extração das características dos sinais pode ser feita utilizando a transformada de wavelet packet. A eficácia do uso deste método para obtenção das características de sinais variantes no tempo é descrita no trabalho de (QIU et al., 2006).

Considerando a forma como os dados foram obtidos, pode-se supor que o atuador opera em três situações distintas: em falha, em funcionamento normal e em teste. O funcionamento em falha se refere à operação utilizando uma das engrenagens com dentes quebrados sem acionamento do freio mecânico. O modo normal é definido como a operação quando é sabido que nenhuma das engrenagens apresenta estado de degradação e o freio mecânico não é utilizado. Por fim, o modo de teste é aquele onde o equipamento está sendo analisado, a fim de determinar o nível de degradação.

Os três tipos de situações definidas pela extração dos dados podem ser utilizadas para treinamento do equipamento. Dessa forma, como apontado em (LAZZARETTI, 2012), uma das formas de se fazer o treinamento do equipamento é utilizando o método da Regressão Logística. Para utilização, o método necessita tanto de dados de funcionamento em falha, como de funcionamento normal. Portanto, a Regressão Logística é empregada.

4.4 Definição dos experimentos

Os experimentos a serem realizados tem por base validar a proposta apresentada neste trabalho. Uma das formas de validação é a verificação do correto funcionamento de todas as entidades após a implementação. Esta validação engloba o teste de comunicação entre as entidades e o correto funcionamento dos serviços implementados.

Outro ponto importante aplicado à validação da arquitetura proposta é a comparação com os métodos tradicionais empregados em trabalhos sobre manutenção inteligente. Conforme apresentado na seção 3.1, os trabalhos relacionados com manutenção inteligente visam a resolução de um problema específico. Dessa forma, a utilização das ferramentas de análise de dados são desenvolvidas ou implementadas especialmente para o caso de uso proposto. Contudo, é possível avaliar o impacto da utilização do sistema proposto para resolução deste tipo de problema.

Além da comparação com os métodos tradicionais, é possível a verificação do comportamento do sistema proposto com o aumento do número de dispositivos monitorados por parte do operador do sistema. Por premissa, pelo sistema apresentar entidades que possibilitam a configuração e análise automática dos dados dos dispositivos, tem-se que o aumento de estações monitoradas não dificulta o processo de análise. Porém, o aumento excessivo pode ocasionar outros problemas, como indisponibilidade da rede ou das entidades de configuração e análise de dados.

4.4.1 Verificação da interoperabilidade entre as entidades

Como forma de validar o correto funcionamento dos componentes da arquitetura proposta, propõe-se um teste de interoperabilidade. O teste visa verificar se todas as entidades estão operando corretamente. Também permite identificar a troca de mensagens entre as entidades e o correto funcionamento dos serviços implementados.

Para validação, o teste deve iniciar e executar uma aplicação mínima. Além da configuração das entidades básicas, como o Gerenciador de Dispositivos e o Gerenciador de Análises, o teste proposto visa validar a análise de um dispositivo. Dessa forma, o dispositivo deve ser corretamente configurado através do Gerenciador de Dispositivos e analisado através do Analisador de Dispositivos. Para a análise, um plano deve ser criado e dados de treinamento enviados.

Com o dispositivo configurado, deve-se verificar os resultados do nível de degradação. O valor de confiança deve ser calculado através da análise dos dados simulados. Os dados são processados pelas ferramentas de análise juntamente com os dados de treinamento.

4.4.2 Comparação da proposta com métodos tradicionais

O experimento visa comparar o método tradicional empregado na análise de degradação de um equipamento com a arquitetura proposta. Nos métodos tradicionais, um software contendo as ferramentas necessárias para a aplicação é utilizado para análise dos dados. Neste caso, é necessário um operador especializado para realizar coleta dos dados, a configuração das ferramentas de análise e a interpretação dos resultados obtidos. Normalmente, em cada nova verificação do coeficiente de degradação, o processo deve ser repetido de forma manual ou semi-automática. Um exemplo deste tipo de software é o Watchdog Agent.

Com a arquitetura proposta e a utilização dos componentes de software apresentados anteriormente, o método de análise dos dados é possível de ser melhorado em comparação com os métodos tradicionais. O experimento visa verificar qual a influência na configuração de uma análise de dados de um equipamento e a forma como os resultados são obtidos.

4.4.3 Verificação da escalabilidade da proposta

Com o aumento do número de dispositivos, utilizando os métodos tradicionais, é possível que exista aumento da dificuldade de monitoração. Dessa forma, o experimento visa determinar qual a influência no aumento do número de equipamentos que necessitam de monitoramento no método tradicional e com a utilização da arquitetura proposta.

Também é objetivo do teste verificar as formas de configuração e definição dos planos de análise para um conjunto de equipamentos. Em métodos tradicionais, para cada novo equipamento que necessita ser testado, os dados precisam ser analisados individualmente pelo operador do sistema. Como forma de melhorar esta condição, o sistema proposto pode apresentar um ganho na configuração e monitoramento de mais de um equipamento de forma simultânea.

5 IMPLEMENTAÇÃO E RESULTADOS

Este capítulo descreve a implementação e resultados obtidos com o sistema proposto. A implementação faz referências diretas sobre o que foi apresentado no capítulo anterior, onde a proposta geral do trabalho foi definida. A forma como as entidades que fazem parte da arquitetura orientada a serviços foram implementadas é descrita e são apresentados resultados.

Alguns experimentos visam comprovar as vantagens na utilização da proposta definida neste estudo. São analisadas questões como a facilidade no gerenciamento das análises e equipamentos, bem como a escalabilidade da solução. Também são testadas as entidades implementadas, verificando a interoperabilidade entre os diferentes softwares desenvolvidos.

5.1 Implementação das entidades da arquitetura proposta

Para a implementação de todas as entidades de software utilizou-se a linguagem de programação Java (JAVA, 2013). Aplicativos construídos utilizando esta linguagem executam em uma máquina virtual. Dessa forma, é possível a execução em diferentes sistemas operacionais sem a necessidade de reescrita ou reimplementação de partes do software. A única premissa é que exista uma versão compatível da máquina virtual para o sistema operacional alvo.

Neste trabalho, a especificação SOA escolhida foi a DPWS. Conforme visto na seção 2.2, como SOA é um padrão que não define a forma de implementação, é possível encontrar diversas. Dentre as várias opções, algumas são voltadas para o cenário corporativo, para a integração entre grandes empresas, e que estão focadas na manipulação e gerência de um grande volume de dados e operações. Outras são empregadas em cenários mais simples do ponto de vista de número de tarefas e conjunto diferenciado de serviços. Dentre as várias especificações, o DPWS foi o mais indicado para este trabalho devido às funcionalidades disponibilizadas, as quais cumprem com os requisitos estipulados no projeto da arquitetura.

A especificação DPWS possui algumas diferentes implementações em meio à di-

ferentes projetos. Dentre os projetos, pode-se citar WS4D, SOA4D e .Net Microframework (MARCELO, 2013). Os dois primeiros contemplam implementações em linguagens Java, C e C++, o que reduz a dependência por um sistema operacional específico. A terceira faz parte de um conjunto de ferramentas desenvolvido pela Microsoft para sistemas embarcados, possuindo implementações em C#. Neste trabalho optou-se pela utilização da implementação Java Multi Edition DPWS Stack (JMEDS), que faz parte do projeto WS4D (WS4D-JMEDS, 2013). Esta escolha partiu do fato de que o projeto em questão conta com grande atividade e contribuidores e, como as entidades de software são implementadas em Java, a integração entre a biblioteca se dá de forma facilitada.

5.1.1 Implementação do Gerenciador de Dispositivos

Como visto na subseção 4.1.3, o Gerenciador de Dispositivos se trata de um componente de software responsável pelo gerenciamento dos dispositivos na arquitetura proposta. De forma geral, o gerenciador é utilizado pelo operador do sistema para obter os dispositivos presentes na rede e configurá-los. A figura 25 ilustra a tela principal do Gerenciador de Dispositivos.

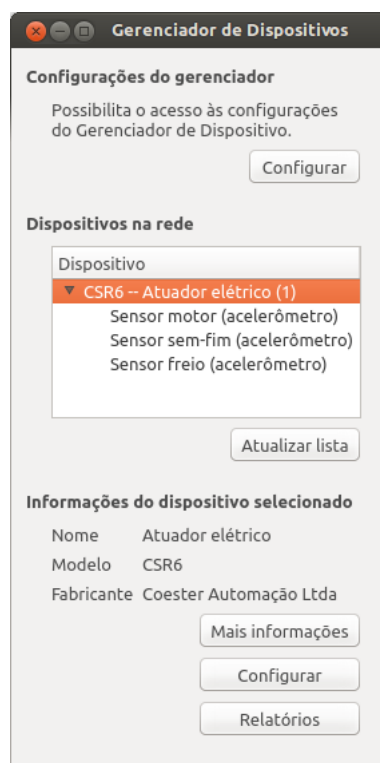


Figura 25: Tela principal do Gerenciador de Dispositivos.

5.1.1.1 Obtenção da lista de dispositivos e serviços

Uma das funcionalidades do gerenciador é a obtenção da lista de dispositivos presentes na rede. O conjunto de ferramentas JMEDS (WS4D-JMEDS, 2013) implementa a

especificação WS- Discovery, permitindo a descoberta de serviços através de um protocolo padronizado. Dessa forma, toda a busca e troca de mensagens para a descoberta e identificação dos serviços é disponibilizada de forma transparente para o desenvolvedor da aplicação.

Com a lista de dispositivos, é possível saber o estado de cada um. Ainda conforme a figura 25, é possível observar que o gerenciador apresenta uma área onde são exibidas as informações pertinentes do dispositivo, como nome, modelo e fabricante do dispositivo. Estas informações são apresentadas quando o operador do sistema seleciona um dos elementos da lista de dispositivos encontrados e referem-se à tabela 1.

O operador tem a possibilidade de visualizar informações mais detalhadas sobre o dispositivo selecionado na tela principal do gerenciador. Esta funcionalidade é disponibilizada através do botão *Mais informações*, localizado na área de informações do dispositivo. Ao ser pressionado, um novo diálogo é aberto. Em complementação às informações do dispositivo, apresentadas na tela principal do gerenciador, o novo diálogo exibe informações mais detalhadas, como uma Uniform Resource Locator (URL) para o número do modelo e outra para o endereço eletrônico do fabricante. A fim de atualizar estas informações, o operador do sistema poderá editá-las utilizando outro botão. Ao ser pressionado, um novo diálogo é aberto, possibilitando a inserção, edição ou atualização de todas as informações do dispositivo selecionado. A figura 26 ilustra a janela de diálogo contendo as informações do dispositivo, a lista de sensores e os botões mencionados anteriormente.



Figura 26: Diálogo com informações detalhadas sobre o dispositivo e lista de sensores disponíveis.

Informações detalhadas sobre cada um dos sensores presente no dispositivo também podem ser obtidas através do Gerenciador de Dispositivos. No diálogo de informações do dispositivo, após a seleção de determinado sensor, os campos à direita são preenchidos com informações adicionais sobre o componente. Esta funcionalidade também é ilustrada na figura 26.

5.1.1.2 Topologia da rede e configuração da base de dados

A topologia da rede é obtida pelo Gerenciador de Dispositivos no momento em que o operador do sistema realiza uma nova busca de dispositivos. Como anteriormente apresentado na figura 25, a tela principal do gerenciador lista todos os dispositivos encontrados, bem como os subdispositivos, localizados em níveis hierárquicos inferiores. Isso é possível pelo fato de que cada dispositivo da arquitetura proposta apresenta um identificador único. O identificador, no caso implementado, uma URL, é utilizada para construir a hierarquia dos dispositivos da rede.

Na URL de identificação de cada dispositivo estão codificadas informações para que o gerenciador possa criar a topologia da rede. Como apresentado na subseção 4.2.1, os dispositivos são codificados seguindo um padrão de nomes, os quais são separados pelo caractere "/". A convenção utilizada facilita a definição de dispositivos que fazem parte de dispositivos com nível hierárquico maior. Também é possível a definição de um nome para a representação de um grupo de dispositivos ou área onde os dispositivos estão localizados.

A configuração da base de dados é fundamental para o correto funcionamento do sistema proposto. A base pode ser configurada através da tela principal do Gerenciador de Dispositivos, pelo principal de configuração do gerenciador, sendo que, após este procedimento, todos os dispositivos serão informados de como o acesso a base deve ser realizado. O gerenciador também a utiliza para manter uma referência dos dispositivos encontrados na rede. Caso um dos dispositivos encontrados não apresentar registro na base de dados, o gerenciador deve criá-lo. Além do dispositivo, também são criados os registros de subdispositivos ou sensores. O diálogo de configuração do Gerenciador de Dispositivos é apresentado na figura 27.

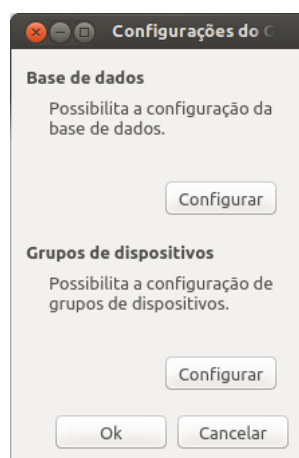


Figura 27: Diálogo de configuração do Gerenciador de Dispositivos.

5.1.1.3 Configuração dos dispositivos

A configuração dos dispositivos é realizada pelo operador do sistema também através do Gerenciador de Dispositivos. A configuração inclui o mapeamento dos sensores, para armazenar os dados de interesse, a definição de comportamentos, que influenciam no modo de operação do equipamento, e o envio de dados de treinamento. As configurações do dispositivo são acessadas pelo diálogo apresentado na figura 26. O diálogo é acessível através da tela principal do Gerenciador de Dispositivos. Para ativá-lo, é necessário que o operador do sistema selecione um dos dispositivos da lista antes do processo de configuração.

No diálogo de configuração, além de outras informações pertinentes ao dispositivo selecionado, estão a lista de sensores que o equipamento possui. As informações de um determinado sensor são exibidas no momento em que o usuário o seleciona na lista de sensores. Dentre as informações, estão o fabricante, modelo e versão do sensor, além de uma descrição e a unidade de medida utilizada na aquisição dos dados. A interface também disponibiliza algumas configurações para o sensor selecionado, como a taxa de amostragem e o número de dados que necessitam ser amostrados para envio à base de dados. Estas configurações estão disponíveis em um outro diálogo, acessível através do botão identificado como *Configurar aquisição*.

A configuração dos comportamentos do equipamento, conforme descrito na subseção 4.1.3, também é feita no Gerenciador de Dispositivos. Os diferentes comportamentos definidos para um equipamento possibilitam que ele opere de diferentes maneiras dependendo do nível de degradação. Os comportamentos configurados estão acessíveis através do diálogo de configuração do dispositivo (figura 26). O diálogo de configuração dos comportamentos é apresentado na figura 28. Ao selecionar um dos elementos da lista, as informações adicionais são preenchidas. O diálogo permite ainda o gerenciamento dos comportamentos, como o envio ou exclusão de um elemento existente. O envio é feito através de um botão, que, quando acionado, abre um novo diálogo para a criação de um comportamento. O operador deverá definir um nome para o novo comportamento e anexar um arquivo contendo o código executável no formato Java Archive (JAR). Dessa forma, o comportamento é enviado para o dispositivo e poderá ser executado em alguma situação, dependendo do tipo de degradação observado.

Os comportamentos seguem uma interface padronizada. A listagem 4, presente no apêndice A, ilustra a interface utilizada para a construção de novos comportamentos. Na implementação do sistema de carregamento dinâmico de códigos executáveis, utilizou-se a biblioteca *jspf*, versão 1.0.2 (JSPF, 2013). A biblioteca permite a definição de *plugins*, o que facilita a implementação das funcionalidades desejadas. Na listagem em questão, são definidos métodos para armazenar o nome (*name*) e a descrição (*description*) do comportamento, além de um método onde a lógica do comportamento deve ser implemen-

tada (*execute*). O método de execução é invocado cada vez que as condições para as quais o comportamento foi configurado forem satisfeitas, o que representa o valor calculado do valor de confiança estar entre os limites estipulados pelo operador do sistema. Em cada invocação do método de execução, o último resultado do valor de confiança é informado, possibilitando que seja utilizado como critério conjunto para a tomada de decisão.

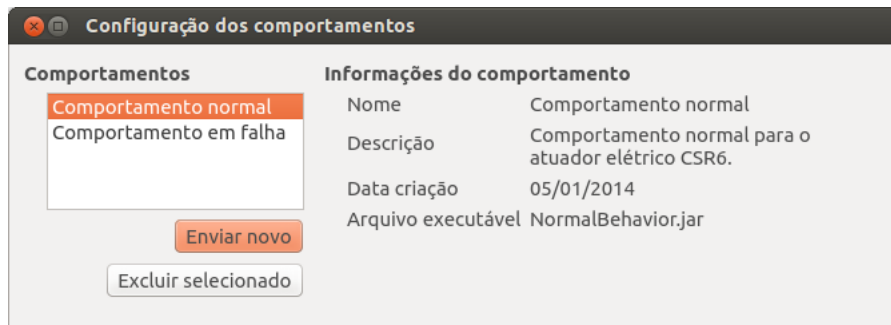


Figura 28: Diálogo para configuração dos comportamentos do dispositivo.

A implementação de um comportamento é apresentada na listagem 5, também presente no apêndice A. O comportamento desta implementação visa que o sistema assuma uma situação, indicada pelo resultado da análise, como normal. A listagem contém a definição do nome, descrição e a ação que será executada pelo comportamento. O método de execução deve conter a lógica necessária para que o dispositivo assuma a condição desejada frente ao valor de confiança obtido através da análise dos dados. Na compilação do arquivo, será gerada uma biblioteca de código no formato JAR. A figura 29 ilustra o diagrama de classes contendo a interface para a criação dos comportamentos e a definição de alguns comportamentos (*NormalBehavior*, *FaultBehavior*, *PreFaultBehavior*). Um arquivo de biblioteca de código no formato JAR é gerado para cada comportamento, os quais podem ser enviados para os dispositivos pelo operador do sistema.

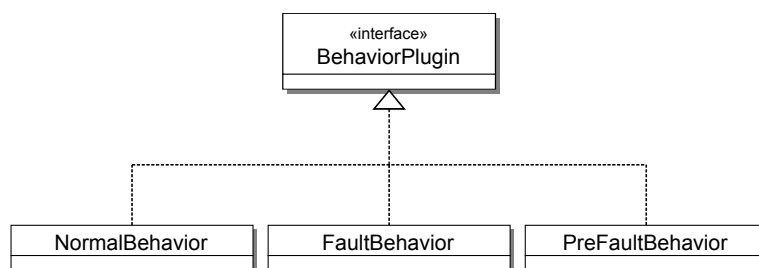


Figura 29: Diagrama de classes da interface para criação de novos comportamentos.

Os dados de treinamento são acessíveis pelo mesmo diálogo de configuração dos dispositivos, conforme figura 26. No diálogo de dados de treinamento, o operador poderá escolher para qual sensor os dados serão mapeados e qual o tipo dos dados, conforme apresentado na figura 30. O tipo de dados pode ser normal ou falha, os quais representam, respectivamente, uma situação em que o equipamento está se comportando conforme

o esperado e quando existe algum comportamento indesejado. Também é necessária a inclusão de informação sobre a origem dos dados, no campo denominado de descrição. O diálogo aceita arquivos de dados no formato Comma-Separated Values (CSV). A figura 30 ilustra a interface de envio de dados de treinamento.

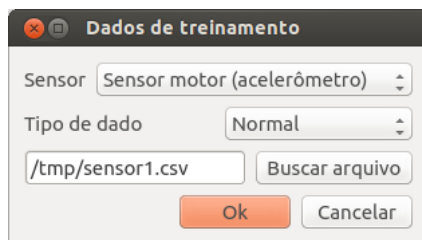


Figura 30: Diálogo de envio de dados de treinamento.

5.1.1.4 Definindo grupos de dispositivos

A configuração ou envio de dados para um conjunto de dispositivos de mesma natureza é possível através da criação de grupos de dispositivos. Dessa forma, dispositivos podem vir a ser agrupados tendo em comum o modelo ou fabricante, por exemplo. A configuração dos grupos é acessível através do diálogo de configuração do Gerenciador de Dispositivos, ilustrado na figura 27. A interface para configuração e gerenciamento dos grupos de dispositivos é apresentada na figura 31. Nela estão presentes a lista de grupos criados pelo operador do sistema e as possibilidades de gerenciamento (criar, editar e/ou excluir). Ao selecionar um grupo existente, na lista à esquerda da figura em questão, os dispositivos que o compõe são apresentados. Pode-se adicionar ou remover dispositivos através da funcionalidade de edição de grupos.

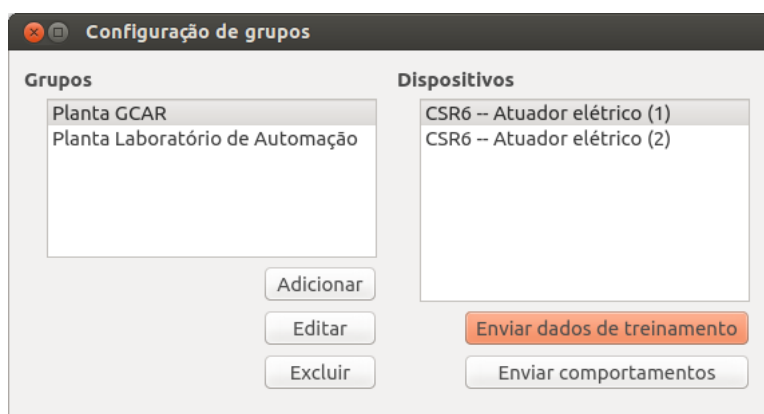


Figura 31: Diálogo de definição e configuração de grupos de dispositivos.

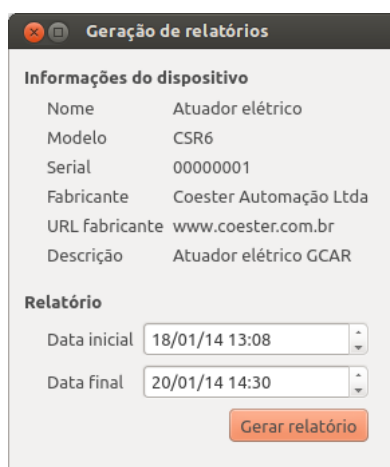
As mesmas operações permitidas para um dispositivo individual são possíveis para os grupos. Sendo assim, é possível a definição de novos comportamentos para o grupo de dispositivos e o envio de dados de treinamento. O envio de novos comportamentos utiliza o mesmo mecanismo apresentado anteriormente, quando abordado do diálogo de confi-

guração de um dispositivo, exceto que, no caso de grupos, um mesmo comportamento é enviado para um conjunto de dispositivos.

O envio de dados de treinamento também segue o mesmo padrão apresentado anteriormente na descrição da configuração de somente um dispositivo. O caso particular se encontra no que diz respeito a escolha do sensor para o qual os novos dados serão mapeados. O Gerenciador de Dispositivos agrupa automaticamente os sensores de mesmo nome, dessa forma, o envio de dados de treinamento é feito com base no nome do sensor. Ao acessar o diálogo de envio de dados de comportamento, os sensores de todos os dispositivos do grupo estarão disponíveis. Será ignorado o sensor escolhido para envio dos dados que não estiver presente em nenhum dos dispositivos.

5.1.1.5 Obtenção de relatórios de saúde

Os relatórios de saúde podem ser obtidos pelo operador do sistema através do diálogo apresentado na figura 32, acessível pela interface de configuração de um dispositivo. O diálogo permite definir o intervalo para geração do relatório, o qual contém os valores de confiança obtidos para cada lote de dados. Os dados são obtidos diretamente da base de dados e o relatório é fornecido ao operador do sistema em arquivo texto no formato CSV.



A imagem mostra uma janela de diálogo intitulada "Geração de relatórios". Ela contém duas seções principais: "Informações do dispositivo" e "Relatório".

Na seção "Informações do dispositivo", há um formulário com os seguintes campos:

| | |
|----------------|------------------------|
| Nome | Atuador elétrico |
| Modelo | CSR6 |
| Serial | 00000001 |
| Fabricante | Coester Automação Ltda |
| URL fabricante | www.coester.com.br |
| Descrição | Atuador elétrico GCAR |

Na seção "Relatório", há dois campos de data:

- Data inicial: 18/01/14 13:08
- Data final: 20/01/14 14:30

Abaixo dos campos de data, há um botão laranja com o texto "Gerar relatório".

Figura 32: Diálogo para obtenção de relatórios de saúde.

5.1.2 Implementação do Gerenciador de Análises

Como apresentado na subseção 4.1.4, o Gerenciador de Análises é um dos componentes de software presente na arquitetura proposta que auxilia no processo de criação de planos de análise para os diversos dispositivos presentes no sistema. Através dele, o operador do sistema pode gerenciar o processo de análise de um equipamento, definindo as ferramentas utilizadas para manipulação dos dados, bem como a origem dos dados utilizados. Além disso, é possível que o equipamento assuma diferentes comportamentos baseado no resultado das análises realizadas. A tela principal do Gerenciador de Análises

é apresentada na figura 33.



Figura 33: Tela principal do Gerenciador de Análises.

As análises cadastradas anteriormente são exibidas em uma lista na tela principal do Gerenciador de Análises. Quando da seleção de um dos elementos da lista, informações adicionais, como nome, descrição e número de dispositivos que utilizam a mesma análise, são fornecidas. Também são fornecidas informações sobre a execução da análise, como a última vez em que o plano de análise foi executado, e se a análise está ativa ou não. As análises e respectivas informações são obtidas diretamente da base de dados.

5.1.2.1 Definição de um plano de análise

O operador do sistema poderá definir novas análises ou gerenciar as existentes, conforme figura 33. Na definição de uma nova análise, o diálogo, ilustrado na figura 34, é aberto. Através do diálogo, o operador deve definir um nome e descrição para o plano de análise, bem como quais dispositivos e sensores farão parte dela. Também deve definir as ferramentas computacionais utilizadas e, se pertinente, configurar quais comportamentos os dispositivos devem assumir frente a diferentes situações de degradação. A periodicidade da execução da análise também pode ser configurada. O diálogo segue as definições encontradas na subseção 4.1.4.

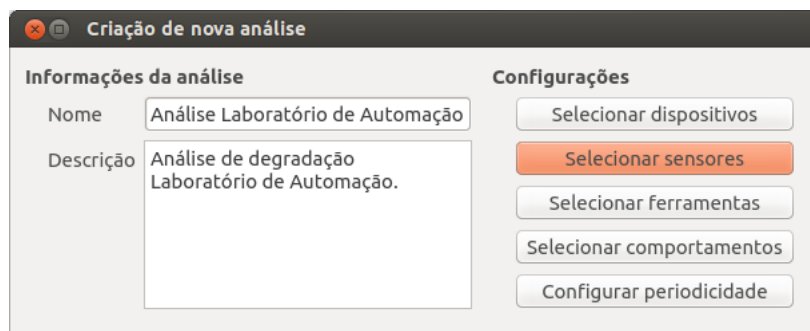


Figura 34: Diálogo de criação de uma nova análise.

A configuração de dispositivos e sensores é realizada em um novo diálogo. O diálogo permite ao operador configurar quais dispositivos serão analisados. Conjuntamente, ao

selecionar os dispositivos, também é possível definir quais sensores serão utilizados na análise de degradação. O princípio de escolha dos dispositivos é herdado do Gerenciador de Dispositivos, exceto por não ser possível a criação de novos grupos de dispositivos através do Gerenciador de Análises. O operador pode optar pela escolha de somente um dispositivo, bem como de um grupo de dispositivos. No entanto, o grupo de dispositivos deve ter sido criado previamente através do Gerenciador de Dispositivos.

As ferramentas de análise dizem respeito aos algoritmos utilizados para analisar os dados obtidos. O processo de escolha é apresentado no diálogo da figura 35. Dois conjuntos de ferramentas estão disponíveis na implementação do Gerenciador de Análises: processamento de sinais e extração de características e avaliação do desempenho do sistema. Transformada rápida de Fourier, análise tempo-frequência e energias da transformada wavelet packet fazem parte do primeiro conjunto. Por conseguinte, regressão logística e reconhecimento estatístico de padrões fazem parte do segundo conjunto de ferramentas.

Se implementado, as ferramentas permitem a configuração dos algoritmos. Após a seleção de uma delas, o operador poderá optar por configurá-la. Para tanto, estão disponíveis botões na interface com o usuário. Ao clicá-los, novos diálogos são abertos com as configurações específicas da ferramenta selecionada.



Figura 35: Diálogo para escolha das ferramentas utilizadas na análise.

O operador poderá também definir os comportamentos utilizados como resposta aos níveis de degradação obtidos. O diálogo da figura 36 ilustra alguns comportamentos e as informações pertinentes àquele que foi selecionado. No procedimento para seleção dos comportamentos utilizados pelo dispositivo, o operador deverá indicar, através das caixas de seleção, quais comportamentos quer utilizar e definir as faixas de variação do valor de confiança onde o algoritmo estará ativo. Como o valor de confiança pode assumir valores no intervalo $[0; 1]$, são disponibilizados dois campos para a seleção do valor mínimo e máximo empregado na definição do comportamento, os quais podem assumir valores dentro desse mesmo intervalo.

A configuração da periodicidade em que a tarefa será executada também pode ser acessada pelo diálogo da figura 34. Um novo diálogo é aberto, onde o operador poderá definir as propriedades de repetição da análise. É possível definir um intervalo fechado,

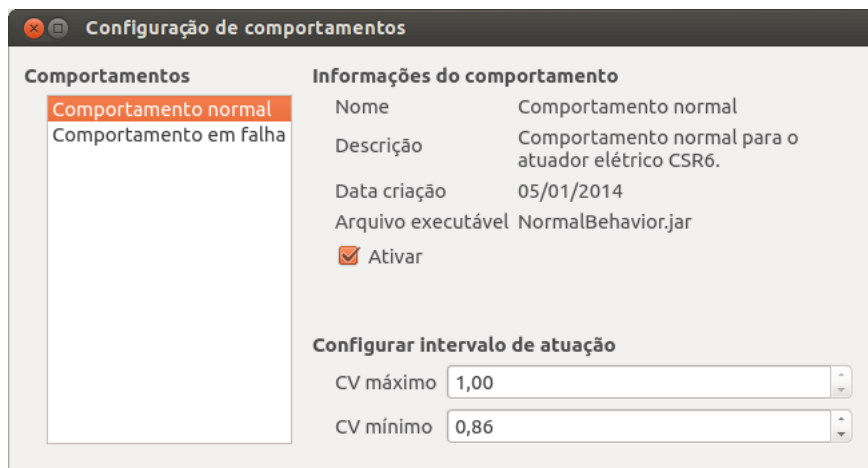


Figura 36: Diálogo para a configuração dos comportamentos dos dispositivos.

onde, após um determinado período, as análises são interrompidas, ou a execução permanente, onde as análises sempre serão executadas. A periodicidade da execução, ou seja, o intervalo entre uma execução e outra, é definido pelo número de amostras coletadas dos sensores. Como apresentado no Gerenciador de Dispositivos, o operador deve definir o tamanho de um lote de dados que será enviado à base de dados. O dispositivo armazena estes dados e, quando o lote estiver completo, o envia a base de dados. A base de dados é monitorada para a execução de uma análise sobre o novo lote de dados, a fim de gerar o novo valor de confiança.

5.1.3 Implementação do Analisador de Dispositivos

O Analisador de Dispositivos é a entidade responsável por executar os planos de análise definidos pelo operador do sistema no Gerenciador de Análises. Conforme definido na subseção 4.1.5, o Analisador de Dispositivos concentra as ferramentas necessárias para análise dos dados e obtenção dos níveis de degradação dos equipamentos. A tela inicial do analisador é apresentada na figura 37.



Figura 37: Tela principal do Analisador de Dispositivos.

O analisador mantém uma fila das análises ativas agendadas pelo operador do sistema.

O Analisador de Dispositivos implementa uma tarefa periódica que verifica a base de dados em busca de novos lotes de dados relativos às análises ativas. Quando novos lotes são encontrados, os respectivos planos de análise são postos em uma fila e executados sequencialmente. Os resultados gerados são armazenados na base de dados.

A tela principal do Analisador de Dispositivos (figura 37) apresenta algumas informações úteis para o operador do sistema. São listados os planos atualmente enfileirados e estatísticas sobre as execuções, como o número de planos agendados e quantas análises estão pendentes para execução. Informações adicionais sobre um determinado plano podem ser obtidas pelo operador através da seleção do elemento na lista da tela principal.

5.1.3.1 Ferramentas para análise dos dados e interoperabilidade entre diferentes fornecedores

As ferramentas de análise dos dados disponíveis no analisador são construídas em forma de interface para os algoritmos presentes no software Watchdog Agent, versão 3.3. Os algoritmos são distribuídos na forma de *scripts* integrados em um software que executa em ambiente MATLAB. A solução empregada no desenvolvimento do analisador foi a de isolar os *scripts* que implementam as ferramentas desejadas e executá-los individualmente através do novo software. Dessa forma, não foi necessária a reescrita dos algoritmos utilizando a linguagem Java.

A execução dos *scripts* MATLAB foi possível utilizando a biblioteca *matlabcontrol* (MATLABCONTROL, 2013), versão 4.1.0. A biblioteca possibilita uma ponte para troca de dados entre um programa escrito em Java e o MATLAB. O mecanismo empregado na solução encontrada é ilustrado na figura 38. Após a coleta dos dados, obtidos da base de dados, para a execução de uma nova análise, o Analisador de Dispositivos seleciona a ferramenta que deve ser empregada, a qual foi definida no plano de análise (1). Como a ferramenta é um *script* Matlab, o analisador carrega os dados e os envia para o Matlab (2), que, por sua vez, repassa-os para os algoritmos presentes no Watchdog Agent (3). Os algoritmos são disparados para execução através da utilização da biblioteca *matlabcontrol* (4). Do mesmo modo, ao final da execução dos algoritmos, o analisador utiliza novamente a biblioteca *matlabcontrol* para recuperar os resultados gerados (5), sendo disponibilizados novamente para acesso no Matlab (6) pelo Analisador de Dispositivos. O processo é repetido se o plano estipular que os resultados anteriores necessitam de uma nova análise. Tanto os resultados intermediários como o resultado final são armazenados novamente na base de dados.

A forma com que o Analisador de Dispositivos foi construído permite a utilização de ferramentas de análise de outros fornecedores. Como exemplo, existe uma versão do Watchdog Agent que executa em conjunto com o software LabVIEW. Devido a uma camada de abstração empregada na separação da definição das ferramentas e na sua implementa-

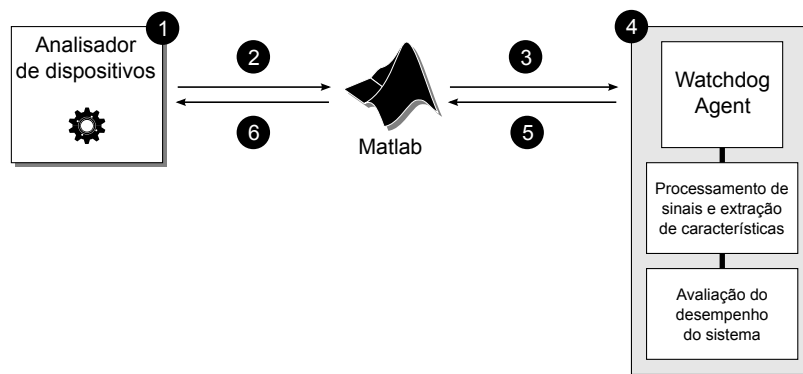


Figura 38: Diagrama de interoperabilidade entre o Analisador de Dispositivos e os *scripts* do Watchdog Agent executando no software Matlab.

ção, é possível a troca ou mesmo a execução simultânea de ferramentas disponibilizadas por diferentes fornecedores, desde que uma nova interface seja provida. A figura 39 ilustra o diagrama de classes utilizado na definição da interface de abstração entre as ferramentas. A interface *IMSTools* é utilizada como base para a construção das ferramentas de análise. Nela, estão definidas todas as interfaces para os algoritmos necessários ao processamento de dados. Já a classe de interface *WatchdogAgent* estende a interface principal provendo uma nova abstração para a implementação das funcionalidades da classe principal utilizando os algoritmos presentes na ferramenta Watchdog Agent. As outras duas subclasses, que utilizam a interface anterior, implementam as ferramentas para manipulação dos dados. Dessa forma, o exemplo ilustra que é possível a extensão do sistema por meio do fornecimento de uma interface para implementação das ferramentas de análise.

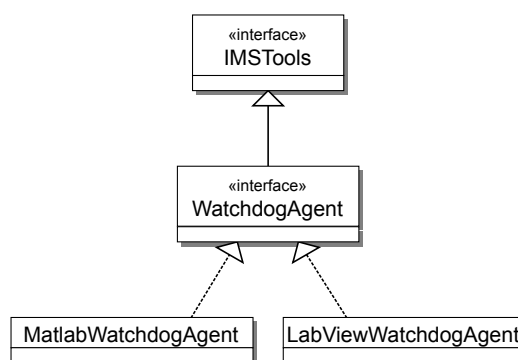


Figura 39: Diagrama de classes para abstração das ferramentas de análise.

5.1.3.2 Alteração do comportamento do dispositivo

Se definido no plano de análise, o dispositivo pode alterar o modo de operação dependendo do nível de degradação obtido com a análise. Como visto anteriormente, o operador do sistema pode definir comportamentos para o dispositivo, a fim de alterar o modo de operação em diferentes situações de degradação. Os comportamentos são definidos através do Gerenciador de Dispositivos e ficam disponíveis no dispositivo configurado.

Após a análise de um plano, o nível de degradação expresso pelo valor de confiança do equipamento é obtido e comparado com as faixas definidas para cada comportamento. Se o valor calculado estiver em alguma das faixas cobertas por um dos comportamentos, o Analisador de Dispositivos automaticamente informa para o dispositivo qual comportamento deve utilizar.

5.1.4 Implementação da camada de acesso a base de dados e definição da estrutura interna

Como apresentado na subseção 4.1.6, é necessária a implementação de uma entidade de software para abstrair o acesso à base de dados através de serviços. O software implementa, para cada componente que necessite de persistência de dados, serviços com operações pertinentes. Por exemplo, a busca ou manipulação de dispositivos pode ser feita com o uso de serviços especializados que manipulam a tabela onde estão armazenadas as informações destes elementos. No entanto, a fim de evitar a redefinição de todas as operações para acesso às diferentes tabelas, o acesso também pode ser feito pelos métodos tradicionais.

Neste trabalho foi utilizada uma base de dados SQLite (SQLITE, 2013). A base de dados SQLite é baseada em arquivos, não necessitando de servidor dedicado, tampouco de um cliente especializado. O conteúdo da base de dados é armazenado em um único arquivo, facilitando a manipulação e distribuição. A escolha por esta base partiu do fato da simplicidade na sua utilização e manutenção, visto que implementa todas as funcionalidades necessárias ao projeto.

Na definição do esquema das tabelas da base de dados, as as informações sobre dispositivos e dados para análise foram separados de modo a facilitar a utilização pelas entidades da arquitetura proposta.

5.1.5 Implementação dos dispositivos

Os dispositivos implementados neste trabalho se referem a atuadores elétricos. Os atuadores, operando em conjunto com válvulas, como comentado na seção 4.3, são do modelo CSR6, fabricados pela empresa Coester. Os equipamentos foram instrumentados e feitas simulações de degradação em diferentes situações. Os dados da simulação das condições foram coletados, sendo possível utilizá-los para simular o comportamento do equipamento sem a necessidade de montagem de novos experimentos físicos em bancada de testes. No caso do embarque físico de um dispositivo no atuador, a técnica utilizada apresenta vantagem, pois o software do dispositivo necessita de modificações somente em pontos específicos, como a forma de obtenção dos dados dos sensores.

Através dos dados dos experimentos armazenados, fez-se necessário o desenvolvimento de uma entidade de software para simular o comportamento do conjunto atuador

elétrico e válvula. A entidade é encarregada de obter os dados prévios e gerá-los nas mesmas condições em que foram obtidos. O software tem por objetivo simular um dispositivo SOA embarcado no equipamento, mas com dados reais, facilitando o teste e validação do sistema. A figura 40 apresenta a tela principal da entidade para simulação dos dispositivos. A tela ilustra algumas informações sobre o dispositivo, como nome, modelo, fabricante e número serial. Estas informações dizem respeito às descritas na tabela 1 e podem ser configuradas pelo operador do sistema. Outras informações gerais, como qual experimento está sendo simulado e as ferramentas utilizadas para a avaliação dos dados, também são apresentadas.

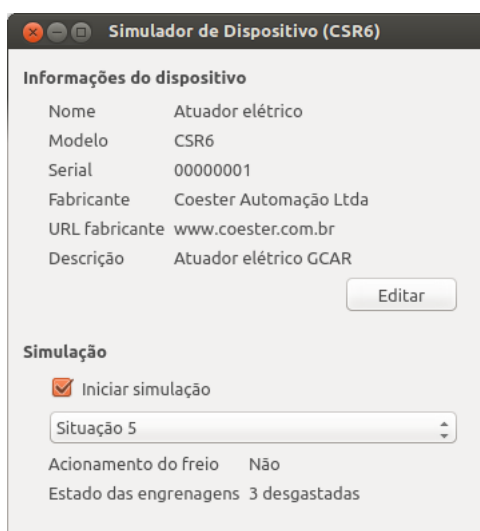


Figura 40: Tela principal do software para simulação dos dispositivos.

Para cada uma das situações apresentadas na tabela 1, estão disponíveis 50 curvas. Desse total, metade são referentes ao fechamento e a outra parte ao processo de abertura da válvula. O dispositivo simula o processo de abertura da válvula indefinidamente, gerando os dados coletados nos experimentos de forma sequencial. Ao término da geração do conjunto de dados, o dispositivo gera os mesmos dados iniciais continuamente.

5.1.5.1 Definição das situações de simulação

As diferentes situações possíveis de simulação estão definidas como apresentado na subseção 4.3.3. De acordo com a descrição do estudo de caso, são possíveis seis experimentos, sendo um deles referente a operação em condições ditas normais, sem interferência no funcionamento, e outras cinco onde o equipamento é submetido a alterações de comportamento. Ao selecionar um dos experimentos, são exibidas informações adicionais, como nome, descrição e as condições a que o equipamento foi submetido. Dentre as condições, estão o acionamento do freio e o uso de engrenagens defeituosas.

5.1.5.2 *Geração dos dados de simulação*

Em cada um dos casos, os dados dos sensores simulados são gerados de acordo com o que foi definido no Gerenciador de Dispositivos. O operador pode definir o número de dados que serão armazenados antes do envio para a base de dados configurada. Neste caso, evitando o acesso contínuo ao serviço de envio de dados da base.

O lote de dados é definido pela quantidade de amostras. Após a aquisição do número de amostras estipulado pelo operador do sistema, o software de simulação agrupa-os em um arquivo CSV. Um arquivo específico para cada sensor é criado. No arquivo, estão discriminados, em colunas, a data e hora de aquisição de cada uma das amostras e o valor amostrado.

5.1.5.3 *Obtenção do valor de confiança*

Como forma de verificação do funcionamento do equipamento simulado, o software disponibiliza informações sobre os valores de confiança obtidos. É possível a visualização do valor de confiança atual, bem como do valor máximo e mínimo obtidos durante o experimento. A data e hora na qual os valores foram gerados também é informada.

O valor de confiança é obtido pela entidade diretamente da base de dados. Dessa forma, fica claro que é possível o próprio equipamento tomar decisões a partir da condição calculada. Em contraste com os comportamentos definidos pelo operador do sistema no Gerenciador de Dispositivos, o equipamento pode apresentar algoritmos padronizados para assumir diferentes comportamentos em situações específicas de degradação.

5.2 Resultados obtidos com a arquitetura proposta

Os resultados obtidos com este trabalho são analisados de forma qualitativa e têm por base as questões levantadas na seção 4.4. De uma forma geral são analisados quesitos como a interoperabilidade das entidades propostas e a comparação do sistema implementado com métodos tradicionais usualmente utilizados. A interoperabilidade entre as ferramentas é analisada, onde testes são realizados para verificar o correto funcionamento das entidades projetadas. Por fim, a proposta é comparada com os métodos tradicionais de análise de degradação, além da verificação das dificuldades e vantagens encontradas na utilização do sistema para monitoramento de um conjunto de equipamentos.

5.2.1 Verificação da interoperabilidade entre as entidades

Em um primeiro momento, todas as entidades foram executadas para verificar a interoperabilidade entre elas conforme sugerido na subseção 4.4.1. O Gerenciador de Dispositivos foi iniciado e colocado em modo de monitoramento. Colocado em modo de monitoração da rede, o gerenciador verifica quando novos dispositivos são iniciados, pos-

sibilitando obter maiores informações sobre cada um.

Após o Gerenciador de Dispositivos, um dispositivo foi iniciado. O dispositivo representa um atuador elétrico. Verificou-se que é possível a alteração das informações do dispositivo, tanto pelo gerenciador, como também pela interface gráfica da entidade de simulação. Este fator aponta o sucesso encontrado na comunicação entre as entidades.

Com o dispositivo configurado e iniciado, criou-se um plano de análise utilizando o Gerenciador de Análises. A análise foi criada obedecendo as condições apresentadas na subseção 4.3.4. Dessa forma, o módulo de processamento de sinais e extração de características foi configurado para utilizar a transformada de wavelet packet, enquanto que, no módulo de avaliação de desempenho do sistema, utilizou-se a ferramenta de regressão logística.

Também foi iniciado o Analisador de Dispositivos. Após a sua inicialização, o plano criado anteriormente é apresentado para execução na lista de planos ativos e agendados. O analisador apresenta as informações detalhadas corretamente, o que novamente comprovou que as entidades se comunicam de forma coerente. Por conseguinte, o analisador foi iniciado, entrando no modo de espera pelos dados do dispositivo.

Com as ferramentas configuradas, foi iniciada a simulação do dispositivo. Utilizando a tela gráfica criada para interface do simulador do atuador elétrico, testou-se as seis situações de teste, conforme apontadas na tabela 3. Como os algoritmos para análise de degradação não foram treinados, o dispositivo não atualizou as informações sobre o valor de confiança calculado em nenhuma das situações testadas.

Para que o dispositivo possa ser analisado corretamente, os algoritmos necessitam passar por um treinamento. Este fato é apontado na subseção 4.3.4. Para o treinamento, foram utilizados alguns dos dados obtidos com os testes nas situações de funcionamento normal e falha. Os dados para treinamento correspondem ao movimento de abertura da válvula. Do total dos dois conjuntos de curvas de abertura, representados por comportamentos normal e em falha, utilizou-se 12 curvas em falha e 12 em funcionamento normal para treinar os algoritmos. Como falha, foram utilizadas as curvas com o pior nível de degradação, obtidas na situação envolvendo uma engrenagem quebrada (tabela 3). Dessa forma, 24 curvas foram utilizadas para o treinamento dos algoritmos.

As curvas de treinamento foram anexadas ao dispositivo através da interface disponibilizada pelo Gerenciador de Dispositivos. Conforme apresentado anteriormente na figura 30, os dados de treinamento podem ser enviados através de uma interface de configuração, sendo possível selecionar para qual sensor os dados serão mapeados e qual o tipo (normal ou falha). Dessa forma, os arquivos de cada uma das curva descritas anteriormente foram enviados e configurados para o respectivo sensor. No experimento, como descrito na subseção 4.3.4, utilizou-se o acelerômetro acoplado ao eixo do motor, o qual registra as variações de vibração.

5.2.2 Comparação da proposta com métodos tradicionais

A proposta de comparação com os métodos tradicionais é apresentada e ilustrada na subseção 4.4.2. São levantadas hipóteses sobre a facilidade de configuração da análise de dispositivos utilizando a arquitetura proposta em comparação com os métodos usuais. Por métodos usuais, entende-se aqueles que se utilizam de um software específico, o qual concentra as ferramentas de análise de dados e é manipulado por operador especializado. Um exemplo deste tipo de software é o Watchdog Agent.

Como anteriormente apresentado nas seções que mostram a implementação do sistema, foram definidos softwares especializados para as tarefas identificadas no processo de obtenção dos níveis de degradação utilizando um sistema de manutenção inteligente. Cada um dos softwares é empregado em uma tarefa distinta na aplicação. Por exemplo, o Analisador de Dispositivos é a entidade que concentra as ferramentas para análise de dados e obtém os níveis de degradação do equipamento analisado. Uma vez que o plano de análise foi criado, o analisador o executa automaticamente conforme planejado. Neste ponto existe a vantagem, em relação ao método tradicional, de que não é mais necessária a obtenção dos dados por parte do operador, tampouco a configuração das ferramentas de análise.

Quanto a aquisição dos dados, o sistema proposto necessita de implementações específicas que não estão disponíveis em todos os equipamentos. No método tradicional, os dados são obtidos diretamente por sensores posicionados estrategicamente pelo fabricante do equipamento ou através de instrumentação posterior por equipe especializada. Com os sensores interligados em placas de aquisição, os dados são amostrados diretamente, sem a necessidade de uma camada de abstração que, no caso da arquitetura proposta, é representada por serviços. Do ponto de vista da aquisição dos dados, os serviços DPWS representam um grande consumo de recursos em um dispositivo. Porém, com a evolução da tecnologia e diminuição dos custos na construção de dispositivos embarcados, o uso de padrões SOA mostra-se viável. Se os dispositivos utilizarem-se desta tecnologia para o fornecimento dos dados, a integração com o sistema de análise será facilitada, o que representa novamente ponto positivo na utilização da arquitetura proposta.

5.2.3 Verificação da escalabilidade da proposta

Este experimento se refere à subseção 4.4.3 e presume que, nos métodos tradicionais, o aumento de dispositivos em um processo de análise de dados fica prejudicado. Também é dada como hipótese o fato da utilização de um mesmo processo de análise para diferentes equipamentos. Utilizando-se os métodos convencionais, a configuração de diversos equipamentos pode representar uma dificuldade.

O Gerenciador de Análises permite a criação de planos que são aplicados para mais de um dispositivo. De certo modo, a análise de mais de um dispositivo pelo método

tradicional não representa grande empecilho. O fator principal é a reconfiguração das ferramentas e injeção de dados diferentes de forma manual a cada nova análise, o que demanda, principalmente, tempo. Mesmo com equipamentos de mesma natureza, a cada nova análise o operador precisa definir manualmente o processo que será executado. Na maioria das vezes, somente é possível a execução dos testes de forma *offline*. A vantagem na utilização da arquitetura proposta reside no fato do processo automatizado criado. Com o Analisador de Dispositivos executando automaticamente os planos, fica evidente a facilidade da análise dos dados. Em adendo, um mesmo plano pode ser empregado para um conjunto de dispositivos, o que, de certo modo, soluciona o problema de reconfiguração das ferramentas e injeção de dados de teste para cada nova análise.

6 CONCLUSÃO

Este trabalho apresentou a proposta de uma arquitetura orientada a serviços para um sistema de manutenção inteligente. Na dissertação foi apresentado o projeto das diversas entidades que fazem parte da arquitetura e propostos alguns casos de uso onde o sistema pode ser empregado. Através dos casos de uso, foi possível determinar as funcionalidades que cada entidade de software deveria apresentar. Por fim, o sistema foi implementado e verificou-se que a solução proposta é viável, visto que integra um ambiente que auxilia na determinação dos coeficientes de degradação de equipamentos de forma facilitada.

Com as entidades propostas, mais especificamente o Gerenciador de Dispositivos, foi apresentado que é possível gerenciar e configurar os dispositivos. É ofertado ao operador do sistema um software dedicado para a configuração dos dispositivos remotamente. O gerenciador possibilita, entre outras funcionalidades, a capacidade de obtenção de todos os dispositivos presentes na rede e a apresentação de informações detalhadas sobre cada um deles ao usuário. Além disso, é possível a obtenção da topologia encontrada na rede. O gerenciador organiza os dispositivos apresentando-os de forma hierárquica, facilitando a verificação da estrutura dos equipamentos. Também no Gerenciador de Dispositivos, é disponibilizada a funcionalidade de obtenção de relatórios de análise dos equipamentos. Dessa forma, o operador pode verificar o histórico de degradação dos equipamentos e, futuramente, utilizar os dados para tarefas de prognóstico.

Como forma de gerenciar as análises empregadas nos dispositivos, foi proposto o Gerenciador de Análises. O gerenciador possibilita a criação de planos de análise para dispositivos individuais ou em grupo. Esta entidade integra todo o processo de gerenciamento das análises a que os equipamentos estão submetidos. É disponibilizado ao operador do sistema a definição das ferramentas que serão utilizadas na análise dos dados, bem como a configuração de comportamentos em função do nível de degradação obtido com a análise. Novamente, como no caso anterior, o Gerenciador de Análises é uma ferramenta de configuração remota, permitindo, assim, o gerenciamento dos planos de análise de forma facilitada.

As análises agendadas pelo operador no Gerenciador de Análises são executadas pela entidade Analisador de Dispositivos. A arquitetura proposta dispõe desta entidade dedi-

cada para a execução dos planos de análise dos equipamentos. O analisador verifica os dados dos dispositivos e executa as análises com base no plano definido pelo operador do sistema. Como apresentado, o analisador utiliza os algoritmos de análise presentes na ferramenta Watchdog Agent. Porém, levando em conta a forma com o que software foi projetado, a extensão do módulo de análise para outras implementações comerciais ou específicas é possível de forma facilitada.

Por fim, como definição do estudo de caso e implementação dos dispositivos, foi possível testar a interoperabilidade da solução proposta. O estudo de caso apresentado foi implementado na forma de dispositivos DPWS. A fim de facilitar os testes da arquitetura, os dados foram obtidos anteriormente em situações distintas. Com isso, o dispositivo se comportou como um simulador, gerando os dados dos sensores na mesma forma como foram obtidos.

Em relação aos experimentos propostos, verificou-se a vantagem no uso da arquitetura proposta em relação a utilização dos métodos tradicionais para verificação da degradação em equipamentos utilizando as técnicas de manutenção inteligente. Ficou claro que, nos métodos tradicionais, utilizando somente o software Watchdog Agent, por exemplo, é necessário o emprego de um operador qualificado para a configuração das ferramentas e obtenção dos resultados. Isso se deve ao fato de que, para cada nova análise, o software precisa ser reconfigurado. Com as hipóteses levantadas, foi possível concluir que a utilização de planos de análise, empregados nesta proposta, facilitam o processo de análise dos dados, visto que, após a criação de um plano, o usuário não necessita modificá-lo nem reconfigurar o software de análise. Também foi constatada a facilidade de obtenção dos níveis de degradação. Ao passo que, nos métodos tradicionais o operador necessita coordenar a análise, com a entidade proposta essas são feitas de maneira automática. O operador do sistema necessita somente obter os relatórios com os valores calculados durante o período estipulado.

As vantagens na utilização da proposta são mais evidentes quando comparadas à configuração e análise de múltiplos equipamentos. As dificuldades encontradas na utilização do software convencional para obtenção dos níveis de degradação de um equipamento são expandidas quando se faz necessária a mesma avaliação para um conjunto de dispositivos. Dessa forma, o operador necessita, a cada nova análise, iniciar todo o processo de configuração das ferramentas e obtenção dos dados. Com a definição dos planos de análise no Gerenciador de Análises, é possível a utilização de um mesmo plano para um grupo de dispositivos. Os dispositivos que fazem parte do grupo de análise serão monitorados de forma igual pelo Analisador de Dispositivos. Dessa forma, o analisador se encarrega de obter os níveis de degradação de forma automática.

Visando a continuidade do desenvolvimento da proposta, pode-se citar alguns pontos verificados como passíveis de melhoramento. Um deles diz respeito aos modos de aquisição de dados de treinamento. No estado atual, o operador do sistema necessita enviar

os dados de treinamento para os dispositivos de forma manual. A cada novo treinamento que deseja-se incluir na base de dados, é necessária a utilização das funcionalidades empregadas no Gerenciador de Dispositivos para envio dos dados. Portanto, com o aumento no número de treinamentos, aumenta também a dificuldade de envio dos dados, por representar tarefas repetitivas. Uma melhoria constatada está na utilização da entidade que representa o dispositivo configurada em um modo específico de aquisição de dados de treinamento. Ao contrário do que é utilizado na implementação atual, o dispositivo poderia ser configurado para enviar os dados das aquisições diretamente para a base de dados como dados de treinamento.

O aumento no número de dispositivos também é considerado um problema na visão do Analisador de Dispositivos. Se muitos dispositivos estão em processo de análise, o analisador manterá uma fila demasiadamente grande. O processamento das análises enfileiradas pode representar atraso considerável na obtenção dos valores de degradação. Considerando isso, a proposta de inclusão de mais entidades para análise de dispositivos pode solucionar o problema. Uma possibilidade é a comunicação ente os processos e verificação da disponibilidade. Dessa forma, evitando que a sobrecarga dos planos de análise agendados recaiam sobre somente um ponto de processamento de dados.

Por último, como forma de melhorias, a reimplementação das ferramentas de análise se faz necessário. Neste estudo, foram utilizadas as ferramentas que integram o software Watchdog Agent. Essas são disponibilizadas em forma de arquivos de *script* para execução no software Matlab. Pelo fato de que arquivos deste tipo são interpretados, verificou-se perda considerável no desempenho em função do aumento do lote de dados analisado. Dessa forma, propõe-se a reimplementação dos algoritmos utilizando linguagens compiladas, como C ou C++, e, em pontos onde for possível, processamento paralelo, como CUDA ou OpenCL (KIRK; HWU, 2012).

REFERÊNCIAS

ANTONINI, M. et al. Image coding using wavelet transform. **IEEE Transactions on Image Processing**, [S.l.], v.1, n.2, p.205–220, 1992.

BELL, M. **Service-Oriented Modeling (SOA): service analysis, design, and architecture**. New Jersey: Wiley, 2008.

BLOCH, H.; GEITNER, F. **Machinery Failure Analysis and Troubleshooting: practical machinery management for process plants**. [S.l.]: Elsevier Science, 2012. (Practical Machinery Management for Process Plants).

BOHN, H.; BOBEK, A.; GOLATOWSKI, F. SIRENA – Service Infrastructure for Real-time Embedded Networked Devices: a service oriented framework for different domains. In: INTERNATIONAL CONFERENCE ON NETWORKING SYSTEMS AND INTERNATIONAL CONFERENCE ON MOBILE COMMUNICATIONS AND LEARNING TECHNOLOGIES, 2006, Mauritius. **Proceedings...** [S.l.: s.n.], 2006. p.43–43.

BÖESCH, K. **Detecção de falhas por fusão de sensores em atuadores elétricos**. 2011. 93p. Projeto de diplomação (Graduação em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2011.

CAMPOS, M. de; TEIXEIRA, H. **Controles típicos de equipamentos e processos industriais**. Rio de Janeiro: Edgard Blücher, 2006.

CANDIDO, G. **Service-Oriented Architecture for device lifecycle support in industrial automation**. 2013. 244p. Tese (Doutorado em Engenharia Eletrotécnica e de Computadores) — Universidade Nova de Lisboa, Portugal, 2013.

CANDIDO, G. et al. SOA at device level in the industrial domain: assessment of OPC UA and DPWS specifications. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS (INDIN), 8., 2010, Osaka. **Proceedings...** [S.l.: s.n.], 2010. p.598–603.

CANDIDO, G. et al. Service-oriented Architecture at device level to support Evolvable Production Systems. In: IEEE INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ELECTRONICS (ISIE), 2010, Bari. **Proceedings...** [S.l.: s.n.], 2010. p.2669–2674.

CANNATA, A.; KARNOUSKOS, S.; TAISCH, M. Dynamic e-Maintenance in the era of SOA-ready device dominated industrial environments. In: KIRITSIS, D. et al. (Ed.). **Engineering Asset Lifecycle Management**. London: Springer, 2010. p.411–419.

CARVAJAL, R. E. G. **Sobre técnicas para manutenção e diagnóstico inteligente de sistemas macetrônicos. Estudo de caso utilizando cálculo de ordem fracionária.** 2011. 219p. Tese (Doutorado em Engenharia Mecânica) — Universidade Estadual de Campinas, Campinas, 2011.

CHOI, J.; NAZARETH, D.; JAIN, H. The impact of SOA implementation strategies on business value and agility: a systems dynamics approach. In: INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION MANAGEMENT AND SERVICE (IMS), 6., 2010, Seoul. **Proceedings...** [S.l.: s.n.], 2010. p.1–6.

CHU, C.; PROTH, J.-M.; WOLFF, P. Predictive maintenance: the one-unit replacement model. **International journal of production economics**, [S.l.], v.54, n.3, p.285–295, 1998.

COLOMBO, A.-W.; KARNOUSKOS, S.; MENDES, J.-M. Factory of the Future: a service-oriented system of modular, dynamic reconfigurable and collaborative systems. In: BENYOUCEF, L.; GRABOT, B. (Ed.). **Artificial Intelligence Techniques for Networked Manufacturing Enterprises Management.** London: Springer, 2010. p.459–481.

DEUGD, S. de et al. Soda: service oriented device architecture. **Pervasive Computing, IEEE**, [S.l.], v.5, n.3, p.94–96, 2006.

DJURDJANOVIC, D.; LEE, J.; NI, J. Watchdog Agentan infotronics-based prognostics approach for product performance degradation assessment and prediction. **Advanced Engineering Informatics**, [S.l.], v.17, n.34, p.109–125, July 2003.

ERL, T. **Service-Oriented Architecture: concepts, technology, and design.** [S.l.]: Pearson Education, 2005.

ESPÍNDOLA, D. **Uma abordagem baseada em modelo para integração e gerenciamento dos dados de sistemas de manutenção inteligente através do uso de técnicas de realidade mista.** 2011. 156p. Tese (Doutorado em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2011.

FACCIN, F. C. **Manutenção Inteligente: fusão de sensores aplicada na detecção de falhas em atuadores elétricos.** 2011. 89p. Projeto de diplomação (Graduação em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2011.

GONÇALVES, L. F. **Desenvolvimento de um sistema de manutenção inteligente embarcado.** 2011. 233p. Tese (Doutorado em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2011.

GONÇALVES, L. F. et al. Design of an Embedded System for the Proactive Maintenance of Electrical Valves. In: ANNUAL SYMPOSIUM ON INTEGRATED CIRCUITS AND SYSTEM DESIGN: CHIP ON THE DUNES, 22., 2009, Natal. **Proceedings...** New York: ACM, 2009. p.8:1–8:6. (SBCCI '09).

GONÇALVES, L. F. et al. Fault Detection, Diagnosis and Prediction in Electrical Valves Using Self-Organizing Maps. **Journal of Electronic Testing**, [S.l.], v.27, n.4, p.551–564, Apr. 2011.

HENG, A. et al. Rotating machinery prognostics: state of the art, challenges and opportunities. **Mechanical Systems and Signal Processing**, [S.l.], v.23, n.3, p.724–739, Apr. 2009.

HOSMER, D.; LEMESHOW, S.; STURDIVANT, R. **Applied Logistic Regression**. [S.l.]: Wiley, 2013. (Wiley Series in Probability and Statistics).

HU, B.; QIN, S. Prognostic methodology for health management of electrical equipments of propulsion system in a type of vessel based on artificial immune algorithm. In: IEEE CONFERENCE ON PROGNOSTICS AND SYSTEM HEALTH MANAGEMENT (PHM), 2012, Beijing. **Proceedings...** [S.l.: s.n.], 2012. p.1–8.

HUA, X.-L.; GONDAL, I.; YAQUB, F. Mobile agent based artificial immune system for machine condition monitoring. In: IEEE CONFERENCE ON INDUSTRIAL ELECTRONICS AND APPLICATIONS (ICIEA), 8., 2013, Melbourne. **Proceedings...** [S.l.: s.n.], 2013. p.108–113.

IMS CENTER. **Documentation of Watchdog Agent Toolbox Algorithms**. [S.l.]: IMS Center, 2007. 61p.

IMS CENTER. **Center for Intelligent Maintenance Systems**. Disponível em: <<http://www.imscenter.net>>. Acesso em: 25 de Fev. de 2014.

JAMMES, F.; SMIT, H. Service-oriented architectures for devices – the SIRENA view. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS (INDIN'05), 3., 2005, Perth. **Proceedings...** [S.l.: s.n.], 2005. p.140–147.

JAVA. **JAVA Programming Language**. Disponível em: <<http://www.java.com>>. Acesso em: 12 de Jun. de 2013.

JOSUTTIS, N. **SOA in Practice: the art of distributed system design**. [S.l.]: O'Reilly Media, 2009.

JSPF. **Java Simple Plugin Framework**. Disponível em: <<https://code.google.com/p/jspf>>. Acesso em: 12 de Jun. de 2013.

KARNOUSKOS, S. et al. Towards an architecture for service-oriented process monitoring and control. In: ANNUAL CONFERENCE ON IEEE INDUSTRIAL ELECTRONICS SOCIETY (IECON), 63., 2010, Glendale. **Proceedings...** [S.l.: s.n.], 2010. p.1385–1391.

KIRK, D.; HWU, W. **Programming Massively Parallel Processors: a hands-on approach**. [S.l.]: Elsevier Science, 2012.

LAZZARETTI, E. P. **Avaliação de desempenho de implementações em hardware e software de algoritmos para aplicações de manutenção inteligente**. 2012. 96p. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

LEE, J. et al. Intelligent prognostics tools and e-maintenance. **Comput. Ind.**, [S.l.], v.57, n.6, p.476–489, Aug. 2006.

LEE, J. et al. Informatics Platform for Designing and Deploying e-Manufacturing Systems. In: WANG, L.; NEE, A. Y. C. (Ed.). **Collaborative Design and Planning for Digital Manufacturing**. London: Springer, 2009. p.1–35.

MALLAT, S. A theory for multiresolution signal decomposition: the wavelet representation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [S.l.], v.11, n.7, p.674–693, 1989.

MARCELO, A. **Análise de desempenho da camada de segurança de aplicações DPWS**. 2013. 77p. Dissertação (Mestrado em Ciências da Computação) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.

MARÇAL, R. F. M.; SUSIN, A. A. Detectando falhas incipientes em máquinas rotativas. **Revista Gestão Industrial**, [S.l.], v.1, n.2, Não paginado, 2005.

MATLABCONTROL. **matlabcontrol – A Java API to interact with MATLAB**. Disponível em: <<https://code.google.com/p/matlabcontrol>>. Acesso em: 13 de Maio de 2013.

MIMOSA. **OSA-CBM 3.3.0: open systems architecture for condition-based maintenance**. Disponível em: <<http://www.mimosa.org/?q=resources/specs/osa-cbm-330>>. Acesso em: 25 de Fev. de 2014.

MORITZ, G. et al. Web services on deeply embedded devices with real-time processing. In: IEEE INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES AND FACTORY AUTOMATION ETFA, 2008, Hamburg. **Proceedings...** [S.l.: s.n.], 2008. p.432–435.

MULLER, A.; CRESPO MARQUEZ, A.; IUNG, B. On the concept of e-maintenance: review and current research. **Reliability Engineering & System Safety**, [S.l.], v.93, n.8, p.1165–1187, Aug. 2008.

MULLER, A.; SUHNER, M.-C.; IUNG, B. Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system. **Reliability Engineering & System Safety**, [S.l.], v.93, n.2, p.234–253, Feb. 2008.

NAGORNY, K. et al. An engineering approach for industrial SoA-based systems of systems. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL TECHNOLOGY (ICIT), 2013, Cape Town. **Proceedings...** [S.l.: s.n.], 2013. p.1956–1961.

NGUYEN, D. Q.; BRAMMER, C.; BAGAJEWICZ, M. New tool for the evaluation of the scheduling of preventive maintenance for chemical process plants. **Industrial & Engineering Chemistry Research**, [S.l.], v.47, n.6, p.1910–1924, 2008.

OLDHAM, J.; JAMES, P.; SHAW, B. Delivering Resource Productivity – The Service Solution. **Green Alliance**, London, Não paginado, 2003.

PAPAZOGLU, M. P.; HEUVEL, W.-J. Service oriented architectures: approaches, technologies and research issues. **The VLDB Journal**, [S.l.], v.16, n.3, p.389–415, Mar. 2007.

PATHAK, J. et al. A Service-Oriented Architecture for Electric Power Transmission System Asset Management. In: GEORGAKOPOULOS, D. et al. (Ed.). **Service-Oriented Computing ICSOC 2006**. Berlin: Springer, 2007. p.26–37. (Lecture Notes in Computer Science).

QIU, H. et al. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. **Journal of Sound and Vibration**, [S.l.], v.289, n.45, p.1066–1090, Feb. 2006.

RAGAVAN, S. V.; KUSNANTO, I. K.; GANAPATHY, V. Service Oriented Framework for Industrial Automation Systems. **Procedia Engineering**, [S.l.], v.41, p.716–723, 2012.

RAMOLLARI, E.; DRANIDIS, D.; SIMONS, A. J. A survey of service oriented development methodologies. In: EUROPEAN YOUNG RESEARCHERS WORKSHOP ON SERVICE ORIENTED COMPUTING, 2., 2007, Leicester. **Proceedings...** [S.l.: s.n.], 2007. p.75.

RIBEIRO, L. et al. A generic communication interface for DPWS-based web services. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS (INDIN), 6., 2008, Daejeon. **Proceedings...** [S.l.: s.n.], 2008. p.762–767.

SHIKARI, B.; SADIWALA, C.; DWIVEDI, R. **Automation in condition based maintenance using vibration analysis**. Bhopal: Department of Mechanical Engineering, Maulana Azad National Institute of Technology, 2004.

SQLITE. **SQLite Home Page**. Disponível em: <<http://www.sqlite.org>>. Acesso em: 15 de Maio de 2013.

STARKE, G.; KUNKEL, T.; HAHN, D. Flexible collaboration and control of heterogeneous mechatronic devices and systems by means of an event-driven, SOA-based automation concept. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL TECHNOLOGY (ICIT), 2013, Cape Town. **Proceedings...** [S.l.: s.n.], 2013. p.1982–1987.

THURSTON, M. An open standard for Web-based condition-based maintenance systems. In: IEEE SYSTEMS READINESS TECHNOLOGY CONFERENCE AUTOTESTCON, 2001, Valley Forge. **Proceedings...** [S.l.: s.n.], 2001. p.401–415.

WS4D-JMEDS. **WS4D-JMEDS Web Services for Devices**. Disponível em: <<http://ws4d.e-technik.uni-rostock.de/jmeds>>. Acesso em: 15 de Jul. de 2013.

ZEEB, E. et al. Service-oriented architectures for embedded systems using devices profile for web services. In: INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS WORKSHOPS AINAW'07, 21., 2007, Niagara Falls. **Proceedings...** [S.l.: s.n.], 2007. v.1, p.956–963.

ZHANG, L.; CAO, Q.; LEE, J. Performance Assessment for a Fleet of Machines Using a Combined Method of Ant-Based Clustering and CMAC. **Advances in Mechanical Engineering**, [S.l.], v.2013, Não paginado, Aug. 2013.

ZHAO, F. et al. SOA-based remote condition monitoring and fault diagnosis system. **The International Journal of Advanced Manufacturing Technology**, [S.l.], v.46, n.9-12, p.1191–1200, Feb. 2010.

ZHAO, Z. et al. Predictive maintenance policy based on process data. **Chemometrics and Intelligent Laboratory Systems**, [S.l.], v.103, n.2, p.137–143, Oct. 2010.

APÊNDICE A: CÓDIGOS UTILIZADOS NA PROPOSTA

Este apêndice apresenta alguns dos códigos utilizados no desenvolvimento da proposta. Algumas informações foram omitidas a fim de destacar somente os pontos importantes da implementação das classes.

Listagem 1: Implementação do serviço para cálculo das energias da transformada wavelet.

```
package ppdws.dpws.services;

// ...

public class WaveletPackageEnergiesService
    extends DefaultService {

    public final static String NAMESPACE =
        DataManipulationDevice.NAMESPACE +
        "/WaveletPackageEnergiesService";

    // ...

    public WaveletPackageEnergiesService() {
        super();

        // Defines an endpoint.
        setServiceId(new URI(NAMESPACE));

        // Defines the data type operation.
        SetDataTypeOperation setDataTypeOperation =
            new SetDataTypeOperation();
        addOperation(setDataTypeOperation);

        // Defines the equipment operation.
        SetEquipmentOperation setEquipmentOperation =
            new SetEquipmentOperation();
```

```

        addOperation(setEquipmentOperation);
    }

    public class SetEquipmentOperation extends Operation {
        // ...
    }

    public class SetDataTypeOperation extends Operation {
        // ...
    }
}

```

Listagem 2: Implementação do dispositivo SOA contendo os serviços para manipulação de dados.

```

public class DataManipulationDevice extends DefaultDevice {

    public final static String NAMESPACE =
        "http://ppdws.org/watchdog/datamanipulation";

    public DataManipulationDevice() {
        super();

        // Adds metadata.
        setPortTypes(new QNameSet(
            new QName("DataManipulationDevice")));

        // Adds device name (name is language specific).
        addFriendlyName("en-US", "Data Manipulation");

        // Adds device manufacture (manufacture is language
        // specific).
        addManufacturer("en-US", "ppdws");

        // Adds model name (model is language specific).
        addModelName("en-US", "...");

        // Creates and adds services.
        FastFourierTransformService ffts =
            new FastFourierTransformService();
        addService(ffts);
    }
}

```

```

TimeFrequencyMomentsService tfms =
    new TimeFrequencyMomentsService();
addService(tfms);

WaveletPackageEnergiesService wpes =
    new WaveletPackageEnergiesService();
addService(wpes);

WaveletPackageMomentsService wpms =
    new WaveletPackageMomentsService();
addService(wpms);
    }
}

```

Listagem 3: Implementação do provedor de serviços para a manipulação de dados.

```

package ppdws.dpws.providers;

// ...

public class DataManipulationServiceProvider {

    public static void main(String[] args) {
        JMEDSFramework.start(args);

        // Creates a new data manipulation device.
        DataManipulationDevice device =
            new DataManipulationDevice();

        // Starts the device.
        try {
            device.start();
        } catch (IOException ex) {
            Logger.getLogger(
                DataManipulationServiceProvider.class
                .getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

Listagem 4: Interface para a construção de novos comportamentos.

```

package ppdws.plugins;

```

```

import net.xeoh.plugins.base.Plugin;

public interface BehaviorPlugin extends Plugin {
    public String name();
    public String description();
    public void execute(double cv);
}

```

Listagem 5: Implementação de um comportamento definido como normal seguindo a interface para construção de comportamentos.

```

package ppdws.electricactuator.csr6.behaviors;

import ppdws.plugins.BehaviorPlugin;
import net.xeoh.plugins.base.annotations.PluginImplementation;

@PluginImplementation
public class NormalBehavior implements BehaviorPlugin {

    @Override
    public String name() {
        return "Comportamento normal";
    }

    @Override
    public String description() {
        return "Comportamento normal para o atuador "
            + "elétrico CSR6.";
    }

    @Override
    public void execute(double cv) {
        /*
         * Código, definido pelo usuário, relativo ao
         * comportamento normal que será executado.
         */
    }
}

```