

Universidade Federal do Rio Grande
Mestrado de Modelagem Computacional

Framework de Acompanhamento
Produto-Serviço (FAPS)

Diogo Paludo de Oliveira

Rio Grande, 16 de Agosto de 2012

Universidade Federal do Rio Grande
Mestrado de Modelagem Computacional

Framework de Acompanhamento
Produto-Serviço (FAPS)

Diogo Paludo de Oliveira

Dissertação de Mestrado de Modelagem
Computacional submetida à avaliação, como
requisito parcial à obtenção do título de Mes-
tre em Modelagem Computacional.

Orientador(a): Prof^a. Dra. Sílvia Silva da Costa Botelho

Co-orientador(a): Prof. Dr. Nelson Lopes Duarte Filho

Rio Grande, 16 de Agosto de 2012

Este trabalho foi analisado e julgado adequado para a obtenção do título de Mestre em Modelagem Computacional e aprovado em sua forma final pelo orientador.

Prof^ª. Dra. Sílvia Silva da Costa Botelho

Banca Examinadora:

Prof^ª. Dra. Sílvia Silva da Costa Botelho

C3 – FURG (Orientador)

Prof. Dr. Nelson Lopes Duarte Filho

C3 – FURG (Co-Orientador)

Prof^ª. Dra. Danúbia Bueno Espíndola

C3 – FURG

Prof. Dr. Floriano Carlos Martins Pires Junior

COPPE – UFRJ

Dedico este trabalho principalmente aos meus pais, que me forneceram todas as oportunidades e ferramentas que precisei para alcançar esta e tantas outras realizações na minha carreira profissional. Dedico também este trabalho à minha namorada, Roberta Siqueira Serrano, que me apoiou em todos os momentos com compreensão e companheirismo, sempre do meu lado, me incentivando e auxiliando quando precisei.

Agradecimentos

Agradeço a minha orientadora, Sílvia Silva da Costa Botelho, por mais uma vez ter confiado a mim a realização deste trabalho, me auxiliando em todos os momentos que necessitei, de maneira compreensiva e dedicada.

Agradeço a minha namorada, Roberta Siqueira Serrano, por me acompanhar por todo este trabalho e suportar todos os problemas junto a mim.

Agradeço os meus pais por possibilitarem que eu chegasse até este ponto da minha carreira acadêmica.

Agradeço também aos amigos e colegas, em especial, Jonata Tyska Carvalho e Edevaldo Braga dos Santos, pelo auxílio e trabalho durante estes anos.

Conteúdo

Lista de Figuras	iii
Lista de Tabelas	iv
Lista de Abreviaturas	v
Resumo	viii
Abstract	ix
1 Introdução	1
1.1 Objetivo	2
1.2 Metodologia	3
1.3 Organização	3
2 Conceituação	5
2.1 Sistemas de Produto-Serviço (PSS)	5
2.2 Sistemas Complexos de Produção (CoPS)	8
2.3 Internet das Coisas (IoT)	12
2.4 Arquitetura Orientada a Serviços (SOA)	14
2.5 Convergindo os conceitos	16
3 <i>Framework</i> de Acompanhamento Produto-Serviço	19
3.1 O conceito de hiperambientes	22
3.2 Uma visão geral da arquitetura	22
3.3 Camada de Serviços	23

<i>CONTEÚDO</i>	ii
3.4 Camada de Comunicação	26
3.5 Camada de Gerenciamento	29
3.6 Camada de Representação	31
4 Aplicando o FAPS na Indústria Naval	35
4.1 A Indústria Naval Brasileira	35
4.2 Apropriando Conceitos CoPS ao Ciclo de Vida de um Navio	37
4.2.1 Utilizando o FAPS no Ciclo de Vida de um Navio	40
4.2.2 Considerações Finais	47
4.3 Estudo de Caso - Estaleiro Rio Grande	48
5 Conclusão	53
5.1 Trabalhos Futuros	55
Bibliografia	56

Lista de Figuras

2.1	Evolução do conceito de PSS	7
3.1	Representação do <i>Framework</i> de Acompanhamento Produto-Serviço em Camadas	21
3.2	Arquitetura final do FAPS.	24
3.3	Estrutura de tabelas do banco de dados MySQL.	30
3.4	Página inicial do sistema <i>web</i> de representação dos dados armazenados no banco de dados do FAPS.	32
3.5	Exemplo de tela de listagem no sistema <i>web</i> de representação dos dados armazenados no FAPS.	32
3.6	Maquete virtual de um estaleiro.	34
4.1	Representação gráfica do Ciclo de Vida de um navio.	38
4.2	Representação da arquitetura do FAPS-Naval.	41
4.3	Visão aérea do Estaleiro Rio Grande.	43
4.4	Testes de dispositivos RFID realizados no Estaleiro Rio Grande.	49
4.5	Maquete virtual utilizada nos testes do FAPS.	50
4.6	Interface <i>web</i> de uma aplicação de representação no FAPS.	51
4.7	Aplicação FAPS simulando interações de ambiente real em ambiente virtual.	52

Lista de Tabelas

2.1	Arquitetura para oferta de solução CoPS Básica, Integrada e Total.	10
2.2	Características importantes de cada conceito utilizado no <i>framework</i>	18

Lista de Abreviaturas

3D Três dimensões

3G *3rd Generation*

AHRS *Attitude Heading Reference System*

BD Banco de Dados

BNDES Banco Nacional de Desenvolvimento Social e Econômico

BoL *Beginning-of-life*

BPM *Business Process Management*

BSD *Berkeley Software Distribution*

C3 Centro de Ciências Computacionais

CAD *Computer-Aided Design*

CAE *Computer-Aided Engineering*

CAM *Computer-Aided Manufacturing*

CATIA *Computer Aided Three-dimensional Interactive Application*

CAVE *CAVE Automatic Virtual Environment*

CoPS *Complex Product Systems*

DELMIA *Digital Enterprise Lean Manufacturing Interactive Application*

DER Diagrama Entidade-Relacionamento

EISA Estaleiro Ilha S.A.

EoL *End-of-life*

EPC *Electronic Product Code*

ERP *Enterprise Resource Planning*

FAPS *Framework de Acompanhamento Produto-Serviço*

FPSO *Floating Production, Storage and Offloading*

FURG Universidade Federal do Rio Grande

GPS *Global Positioning System*

HTML *HyperText Markup Language*

IBM *International Business Machines*

IMU *Inertial Measure Unit*

IoT *Internet of Things*

KPI *Key Performance Indicators*

LED *Light-Emitting Diode*

MoL *Middle-of-life*

MVC *Model-View-Controller*

PDI Pesquisa Desenvolvimento Inovação

PHP *PHP: Hypertext Preprocessor*

PLM *Product Lifecycle Management*

PSS *Product-Service Systems*

RFID *Radio-Frequency IDentification*

ROS *Robot Operating System*

SAP Sistema de Acompanhamento de Projetos

SGBD Sistema de Gerenciamento de Banco de Dados

SMS Segurança, Meio ambiente e Saúde

SOA *Service-Oriented Architecture*

TI Tecnologia da Informação

UID *Unique Identification Number*

WBS *Work Breakdown Structure*

XML *eXtensible Markup Language*

Resumo

Atualmente, muitas empresas e indústrias de setores variados se deparam com dificuldades para desenvolver soluções integradas que auxiliem na produção e na relação com os consumidores. A complexidade de tais soluções é cada vez maior, envolvendo diversas áreas que vão além do tradicional comércio de produto como produto final, e arquitetar maneiras de atender o mercado de maneira completa e diferenciada tem sido o principal desafio de grandes empresas.

Os Sistemas Complexos de Produção (CoPS) buscam fornecer uma solução completa e integrada ao cliente. Nele, uma empresa não oferece simplesmente um produto, mas entrega uma solução completa e autônoma que da maneira que o consumidor necessita. Sistemas de Produto-Serviço (PSS), por outro lado, buscam se tornar parte dos processos do cliente, sem o foco em grandes sistemas integrados. Com esta abordagem, o consumidor fornece *feedback* de suas necessidades, fazendo com que a empresa prestadora consiga suprir o mercado com a real necessidade existente, agregando valor rapidamente com soluções cada vez mais completas.

Este trabalho propõe a modelagem de um *framework* computacional capaz de identificar o fluxo de informações em um CoPS, construindo um sistema integrado, que somado aos conceitos de PSS, mostrará uma visão completa dos dados pertinentes a cada componente do sistema. Através do conceito de Internet das Coisas, o *framework* proposto busca integrar um grande número de dispositivos e atores em uma única solução, fazendo com que os componentes se comuniquem e troquem informações dentro da plataforma.

Assim, com o *framework* proposto e conceituado, este será aplicado na indústria naval, mostrando quais setores internos da indústria serão integrados, quais os ganhos na comunicação e conceituando o ciclo de vida do navio, desde a sua montagem e edificação, até o uso e desmonte.

Abstract

Currently, many companies and industries from various sectors have difficulties to develop integrated solutions that help in the production and relationship with consumers. The complexity of such solutions is growing, involving several areas that go beyond the traditional trade in goods as final product, and to plan ways to meet the market thoroughly and differentiated has been the main challenge for large companies.

The Complex Product Systems (CoPS) seek to provide a complete and integrated solution to the customer. In it, a company does not offer just one product, but delivers a complete and autonomous solution in the way that the consumer needs. Product-Service Systems (PSS) on the other hand, become part of the client processes, without focus in large integrated systems. With this approach, the consumer provides feedback on their needs, making the provider company able to supply the market with real existing need, adding value quickly with more complete solutions.

This paper proposes the modeling of a computational framework capable of identifying the flow of information in a CoPS, building an integrated system, which added to the concepts of PSS, show a complete view of the data pertinent to each system component. Through the concept of Internet of Things, the proposed framework seeks to integrate a large number of devices and actors in a single solution, causing the components to communicate and exchange information within the platform.

Thus, with the framework proposed and conceptualized, it will be applied in the marine industry, showing domestic sectors of the industry that will be integrated, which are the gains in communication and conceptualizing the life cycle of a ship, since its assembly and erection, to the use and dismount.

Capítulo 1

Introdução

Os Sistemas de Produto-Serviço (PSS) e os Sistemas Complexos de Produção (CoPS) não são novidade em diversas áreas do mercado, mas parecem encontrar dificuldades em algumas indústrias mais tradicionais. Por exigir uma mudança drástica nos processos internos e na forma de visualizar as atividades realizadas na prestação de um serviço ou no desenvolvimento de um produto, muitas empresas decidem não optar pelas novas abordagens, que podem trazer grandes vantagens quando corretamente aplicadas e bem estruturadas.

Atualmente, com o rápido crescimento da quantidade de dispositivos disponíveis para auxiliar tanto o homem comum quanto empresas e indústria em seus processos, são necessárias mudanças nas maneiras de se visualizar uma solução empresarial, para modernizar e aproveitar a tecnologia disponível no mercado. As ideias de Computação Ubíqua e Internet das Coisas buscam fornecer meios para que diversos componentes, sejam eles *Smartphones*, dispositivos de radiofrequência (RFID) ou sensores, comuniquem entre si compartilhando informações e dados pertinentes ao sistema como um todo. Com a definição das rotinas lógicas do processo da empresa, é possível que informações sejam adquiridas pelo sistema de maneira autônoma, sem a interação homem-máquina.

A união destes conceitos em um único *framework*, capaz de abstrair o fluxo de informações existente entre os componentes e atores de um sistema, tem potencial para unificar processos internos nas empresas, integrando as atividades desempenhadas nos diferentes projetos em um único ambiente virtual. Assim, são evitados retrabalhos de alimentação dos dados em sistemas independentes que não se comunicam, inconsistências

de informações em diferentes bases e, principalmente, se faz possível o uso de todos estes dados coletados constantemente pela aplicação para realimentar os processos, causando melhorias com base nas experiências passadas.

No cenário local, de grandes investimentos e desenvolvimento da indústria naval brasileira, destacando a região sul do Rio Grande do Sul, na cidade de Rio Grande, surge um grande estudo de caso que se encaixa perfeitamente na proposta de sistema apresentada neste trabalho. A indústria naval nacional precisa se modernizar para alcançar níveis internacionais, agilizar processos e reduzir tempo de produção de embarcações. A computação pode oferecer soluções que auxiliem este objetivo.

1.1 Objetivo

O objetivo geral desta dissertação de mestrado é modelar a estrutura de um *framework* computacional capaz de tratar e armazenar as diversas informações e dados que são utilizados na execução de um projeto, unindo em uma solução integrada capaz de se atualizar e retroalimentar.

Especificamente, os objetivos são:

- Pesquisa e determinação do papel da computação em Sistemas de Produto-Serviço (PSS) e Sistemas Complexos de Produção (CoPS)
- Abstrair computacionalmente o fluxo de informações entre diferentes componentes existentes na execução de um projeto realizado em ambiente empresarial/industrial.
- Aplicar o conceito de Computação Ubíqua e Internet das Coisas para universalizar a comunicação entre diferentes componentes que possam participar dos processos.
- Analisar o caso de uso da indústria naval nacional, identificando informações existentes e pertinentes nos processos, setores participantes do ciclo de vida naval e serviços prestados.
- Desenvolver um protótipo da aplicação.

1.2 Metodologia

- Revisão bibliográfica: Fase de pesquisa sobre sistemas utilizados atualmente nas empresas e indústrias de manufatura, construção e engenharia, principalmente de grande porte.
- Definir conceitos pertinentes na implementação: Definir quais conceitos são indicados e pertinentes do ponto de vista acadêmico para enriquecer a pesquisa e proposta de dissertação apresentada neste trabalho.
- Projetar o sistema computacional: Determinar as estruturas necessárias para o funcionamento da solução integrada, utilizando técnicas da Engenharia de Software para modelar o sistema e linguagens de programação pertinentes aos objetivos finais almejados.
- Pesquisar necessidades e testar viabilidade de aplicação na indústria naval: Consultar com profissionais da área naval a viabilidade da aplicação de um sistema com tal proposta na indústria naval. Pesquisar características específicas dos processos de projeto, edificação e uso de um navio, para assim determinar informações que devem ser levadas em consideração para formar um sistema unificado. Testes de desempenho dos dispositivos que serão utilizados na aplicação também serão realizados.
- Desenvolver protótipo: Assim que os conceitos, componentes e estruturas do sistema estiverem devidamente determinados, se inicia a fase de desenvolvimento do sistema, aplicado na indústria naval.

1.3 Organização

Esta proposta de qualificação será dividida em 5 capítulos, onde o primeiro apresenta a Introdução, com Objetivo, Metodologia e Cronograma da dissertação aqui proposta.

O segundo capítulo, Conceituação, apresenta os conceitos pesquisados e pertinentes para a implementação do *framework* proposto no trabalho. Cada conceito terá uma sessão independente que o descreve com base no estado da arte.

O terceiro capítulo, Framework de Acompanhamento Produto-Serviço, apresentará o detalhamento do *framework*, especificando características de funcionamento, estrutura e aplicação dos conceitos apresentados no capítulo 2.

O quarto capítulo, Indústria Naval, mostrará o cenário atual na indústria naval e apresentará os principais processos realizados em estaleiros e na vida útil de um navio, assim como as deficiências na área da computação que poderiam ser corrigidas.

O quinto capítulo, Proposta de Trabalho, apresentará o trabalho proposto em detalhes, mostrando a aplicação dos diversos conceitos utilizados, seu valor acadêmico e sua utilidade de aplicação real no mercado.

Capítulo 2

Conceituação

2.1 Sistemas de Produto-Serviço (PSS)

A comercialização unicamente de produtos não é vista mais como rentável e nem mesmo atrativa ao consumidor. Atualmente, com o objetivo de agregar valor em suas ofertas, empresas e indústrias buscam diferentes maneiras de competir no mercado, por vezes apresentando alternativas que vão além do corte de custos para baixar preços no produto final. Muitas vezes, a alternativa utilizada é fornecer serviços ligados aos produtos oferecidos, com objetivo de melhorar a experiência que o consumidor possui com o produto, tornando a relação empresa-cliente mais atrativa. Unindo o produto com serviços que o complementem e ofereçam um diferencial que, aos olhos do cliente, seja a vantagem em relação aos concorrentes, uma empresa consegue fornecer uma solução mais completa, estabelecendo laços entre os clientes e seus serviços, mantendo uma relação mais próxima das demandas e exigências existentes no mercado ([Mont, 2002]; [Aurich et al., 2006]; [Baines et al., 2007];[Gebauer, 2008]).

Em contraponto ao comércio tradicional, onde vínculo entre consumidor e empresa é desfeito no momento da entrega do produto final, as recentes abordagens de Sistemas de Produto-Serviço (*Product-Service Systems* - PSS) propõem a estruturação de uma solução completa a qual objetiva estender este vínculo ao longo de todo o ciclo de vida do produto, tornando possível a obtenção de um *feedback* rápido e preciso dos problemas que estão ocorrendo na solução proposta. Através de diagnósticos previamente arquitetados e controlados por uma equipe remota ou presencial, problemas são percebidos rapidamente,

podendo ser consertados até mesmo antes de resultados ruins ocorrerem. Portanto, o PSS faz com que a distinção que associava mercadorias há algo tangível e serviços há algo intangível desapareça. Abordagens PSS envolvem todo o ciclo de vida do produto/serviço, envolvendo diferentes profissionais e áreas, requerendo tratamentos de natureza interdisciplinar. Por exemplo, neste novo paradigma surge a Engenharia de Serviço, a qual destaca-se pelo desenvolvimento e projeto sistemático de soluções produto-serviço, utilizando um conjunto adequado de modelos, métodos e ferramentas ([Pezzotta, 2010]).

Um PSS é uma oferta de produto e serviço integrados, que fornece valor agregado no uso da solução. Um PSS oferece a oportunidade de separar o sucesso econômico da aquisição material, reduzindo assim o impacto ambiental da atividade econômica e alterando as relações previamente estabelecidas de propriedade sobre um bem. A lógica PSS tem premissas na utilização do conhecimento do projetista e do construtor para aumentar o valor agregado final e diminuir custos materiais iniciais nos processos. ([Baines et al., 2007])

Exemplos de abordagens PSS podem ser vistos no segmento de TI. *Tablets* e *Smartphones*, diferente de como acontecia há alguns anos atrás, não são mais vendidos apenas com funcionalidades de fábrica. Hoje em dia, empresas adicionam ao pacote serviços atraentes ao consumidor e novas funcionalidades capazes de serem atualizadas quase que em tempo real com o lançamento. Mas as mudanças não são apenas quanto à quantidade de serviços oferecidos junto de um produto, mas também no próprio conceito de propriedade de um bem. Aparece então a possibilidade de oferecer um sistema sem a necessidade de sua compra por parte do usuário. O fornecedor de TI permite o uso de um sistema que desempenhe alguma finalidade chave, seja na administração, operação ou outras áreas de uma empresa, enquanto o cliente paga pelo uso, sem ter qualquer direito de propriedade sobre o sistema em si, apenas dos dados inseridos.

Grandes indústrias têm modificado suas ofertas, buscando nos Sistemas de Produto-Serviço a base para alcançar melhores resultados comerciais e ambientais. Um caso bastante citado na literatura, apresentado por ([Baines et al., 2007]), é o da *Rolls-Royce* (RR) e seu pacote *Total-Care* oferecido às companhias aéreas. Neste, ao invés de transferir a propriedade do motor de turbina a gás para a companhia aérea, a *Rolls-Royce* cobra pelo uso. A sua tecnologia de turbina a gás é líder mundial e suas peças de reposição, assim como os serviços de manutenção oferecidos, são exemplares. Além disso, como a empresa

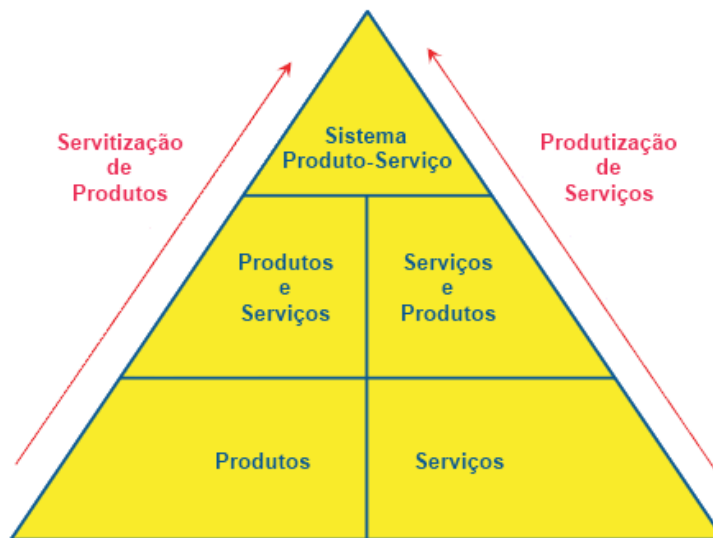


Figura 2.1: Evolução do conceito de PSS

mantém acesso direto ao ativo, podem ser coletados dados sobre desempenho do produto e seu uso. Esses dados permitem melhorias de parâmetros de desempenho (por exemplo, programas de manutenção, etc) para melhorar a eficiência do motor e a utilização dos equipamentos, reduzindo assim os custos totais e o impacto ambiental.

Por outro lado, hoje na indústria, a concepção de um Sistema Produto-Serviço não basta para atender as necessidades existentes. A literatura, quando sobre PSS, foca bastante em seu papel sustentável, destacando ganhos e melhorias que não necessariamente buscam a redução dos custos e aumento de produtividade. Em uma empresa, sustentabilidade não basta para se manter no mercado e crescer financeiramente. Principalmente na indústria naval, onde a quantidade de produtos se limita a milhares, centenas ou até mesmo dezenas em um ano inteiro, é necessária uma abordagem que busque trazer ganho no processo, diminuindo custos agilizando o processo ou mesmo diminuindo o número de hora-homem investido. O trabalho de ([Tukker, 2004]), conclui que não se pode esperar que a maioria dos tipos de PSS resulte em ganhos radicais. Além disso, através de análises para casos de negócios e sustentabilidade utilizando PSS, se percebe a existência de contradições interessantes em relação ao que seria desejável de um ponto de vista sustentável e o que seria adequado sob o ponto de vista de negócio.

2.2 Sistemas Complexos de Produção (CoPS)

Para criar uma solução integrada que fuja da lógica tradicional de manufatura e comercialização, e que consiga atender, além do mercado de varejo e a atacado, a indústria metal-mecânica, civil, naval, entre outras, se faz necessária uma mudança de perspectiva. Nesta, a produção não seria mais baseada em projetos individuais, mas enfatizaria a continuidade e o *feedback* da execução anterior [Cova and Salle, 2007]. Um projeto não seria visto mais com etapas de início, meio e fim, sendo substituído por outro ao final do processo: seria criado um ciclo, onde todos os erros e acertos, além de melhorias de processo e abordagens bem sucedidas, realizadas ao longo do trabalho, serviriam de *feedback* na realimentação ao longo de suas etapas. Nesta nova perspectiva, teria-se como objetivo principal a melhoria dos processos, desde a compra de materiais e edificação, até ao desmonte final, registrando-se e tratando-se as diversas informações pertinentes aos diferentes estágios do ciclo do produto. Surgem assim, os Sistemas Complexos de Produção (CoPS), os quais buscam implementar esta perspectiva, através da integração e constante atualização dos processos.

Segundo [Ying-Tao and Khim-Teck, 2006], CoPS são produtos, sistemas e construções de alto custo e de engenharia intensiva. São bens de capital de negócio para negócio que formam a espinha dorsal da economia moderna e da sociedade. Tais bens podem ir desde grandes prédios inteligentes até sistemas balísticos de precisão. [Davies and Brady, 2000] defendem que empresas podem alcançar o que chamam de “economias de repetição”, instaurando mudanças organizacionais, rotinas e processos de aprendizagem para executar um número crescente de projetos a menor custo e mais eficientes. Ao contrário de investir em projetos únicos, empresas podem oferecer “soluções repetidas”, reciclando experiências de uma execução ou projeto para outras na mesma linha de negócio.

Sistemas Complexos de Produção possuem características que os diferenciam dos modelos tradicionais de produção. CoPS podem ser descritos como:

- bens de custo alto compostos de elementos interconectados, como unidades de controle, sub-sistemas e componentes, que geralmente estão organizados de maneira hierárquica e desenvolvidos para consumidores ou mercados específicos [Hobday et al., 2000]. A principal contribuição destes sistemas reside na concep-

ção do sistema, além do gerenciamento e integração dos projetos, deixando de focar no processo de manufatura;

- sistemas que tendem a exibir propriedades não lineares ao longo do tempo, no sentido que de geração em geração do sistema, pequenas mudanças em uma parte da estrutura podem levar a uma alteração maior em todo o funcionamento. ([Hobday and Rush, 2001]). Propriedades não-lineares dificultam a previsão de problemas. A interação e comunicação entre componentes durante o processo de design, desenvolvimento e integração do sistema são de fundamental importância;
- produzidos em projetos ou pequenas etapas que permitem um alto nível de interatividade dos usuários, através dos quais suas necessidades comerciais alimentam o processo de inovação diretamente. Outra consequência da produção em pequenas etapas é que a definição, desenvolvimento, simulação, teste e produção do produto geralmente envolve transferência de conhecimento dentro da complexa rede de fornecedores e inclui um grande número de interações entre diferentes especialistas ([Heighes, 1997]). Mesmo após entrega e instalação, alguns sistemas ainda necessitam ser revisados e ajustados de acordo com os requerimentos do usuário.

As soluções propostas pelos sistemas denominados como CoPS possuem características diferentes dependendo da necessidade do cliente e do grau de integração com os processos já existentes na lógica de trabalho da empresa. A solução proposta por um sistema complexo abrange tanto aplicações *standalone*, sem a integração com outros processos, passando por abordagens de solução integrada, com sistemas, dispositivos e características específicas de cada cliente, e, por último, soluções estendidas e completas, com funcionalidades inexistentes por padrão. Desta forma, além de apenas oferecer uma solução, a empresa responsável pelo CoPS adquire diferenciais competitivos no mercado, como soluções que se adaptam a lógica do cliente e que não são engessadas e limitadas a uma única lógica de funcionamento.

[Brax and Jonsson, 2009] utiliza a Tabela 2.1 para demonstrar os diferentes tipos de solução e alguns principais aspectos que podem estar presentes em aplicações dos sistemas complexos de produção. Nela são detalhados alguns dos possíveis serviços existentes em um ambiente industrial, onde pacotes de diferentes complexidades são oferecidos ao

Solução Básica	Camada de Customização	Camada de Melhorias
Base instalada: Bens Motores Sistemas auxiliares Equipamentos	Configuração de equipamentos única para cada cliente	Sensores correspondentes
Plataforma da solução <i>Hardware</i> Planejamento de manutenção Histórico de manutenção Gerenciamento de inventário	Ajustar o sistema aos sistemas de TI do cliente	Sistema de Monitoramento (Sistema comprado)
Gerenciamento da informação Agendamento de manutenções recomendadas Instruções de trabalho Dados dos produtos	Projeto do programa de manu- tenção Selecionar dados e instruções dos produtos	Dados em tempo real so- bre condições da execução Relatórios de controle mensais
Componentes de serviço Entrega e instalação do sistema Treinamento dos usuários	Adaptação do conteúdo do sis- tema Suporte	Monitoramento remoto Alertas Consultoria

Tabela 2.1: Arquitetura para oferta de solução CoPS Básica, Integrada e Total.

cliente. Na tabela, as linhas representam as diferentes áreas que a solução, que pode abranger desde puramente equipamentos até mudanças de funcionamento de processos internos e treinamento de pessoal. Os exemplos mostrados são:

- Equipamentos base - Área que se responsabiliza por componentes físicos individuais que não necessariamente fazem parte de um sistema integrado, como motores, sistemas auxiliares e equipamentos em geral;
- Plataforma da solução - Oferece a estrutura necessária para que o CoPS seja utilizado, incluindo o hardware necessário, o planejamento e histórico de manutenção, o gerenciamento de inventário e etc.;
- Gerenciamento da informação - Parte que torna capaz a manipulação dos dados adquiridos pelos equipamentos, sistemas e componentes do sistema, permitindo que relatórios de desempenho, instruções de trabalho e condições de trabalho avaliadas em tempo real;
- Componentes de serviço - Serviços adicionais que podem existir dependendo do uso ou não de outras áreas do CoPS. Se o gerenciamento da informação for utilizado, a personalização do sistema pode ser um serviço oferecido, assim como quando a plataforma de sistema está presente, suporte e consultoria na utilização podem ser de interesse do cliente.

Já as colunas representam o nível de customização possível para a solução CoPS. Na esquerda há uma solução básica, com os requisitos mínimos para que a solução traga benefícios ao cliente. A coluna do meio, representando uma camada de customização, oferece personalizações ao negócio do cliente que podem trazer novos benefícios e melhorias nos serviços desempenhados. Por fim, a última coluna, da direita, denominada camada de melhoria, oferece serviços de atenção especial da empresa prestadora da solução, com consultoria, monitoramento e acompanhamento em tempo real, objetivando ampliar os benefícios e as vantagens que a solução proporciona.

2.3 Internet das Coisas (IoT)

A computação ubíqua (ou pervasiva) é uma solução para a adoção de conceitos de Engenharia de Serviço em um sistema computacional, do ponto de vista da interação e comunicação entre componentes. A importância da computação ubíqua, neste contexto, surge com a afirmação de [Weiser, 1999]: “as tecnologias mais complexas são aquelas que desaparecem...” A computação ubíqua oferece um ambiente integrado para o usuário, onde a percepção de estar utilizando um computador é mínima.

Estudos têm sido realizados visando a criação de ambientes inteligentes, também chamados de ambientes sensíveis ao contexto [Schilit et al., 1994]. Tais ambientes são baseados nas idéias lançadas por *Weiser*, envolvendo computação ubíqua. Na sua proposta, a computação deve estar presente em qualquer momento e em qualquer lugar [Weiser, 1999]. Neste paradigma, diferentes objetos, equipados com computação embarcada, podem integrar uns com os outros, sentindo e se adaptando ao ambiente de uma forma transparente, tornando a interação humano-computador mais simples.

Ambientes inteligentes podem ser definidos como sistemas distribuídos heterogêneos de sensores e atuadores, incluindo serviços de apresentação multimídia, automação e controle predial, objetos físicos inteligentes, nodos de rede com sensores sem fio, dispositivos nômades pessoais ou compartilhados, e muitos outros sistemas e entidades.

Para permitir uma combinação de computação ubíqua com a necessidade de incorporação de tecnologias de rastreamento e aquisição de informações para implantação de uma solução integrada, a Internet das Coisas (IoT) emerge como potencial solução. A IoT é um novo paradigma que torna possível a presença generalizada ao nosso redor de uma variedade de coisas ou objetos que fornecem algum tipo de informação ao ambiente. Alguns dispositivos, como *Radio-Frequency Identification* (RFID), sensores, atuadores, telefones celulares, e assim por diante, permitem a interação e a cooperação entre as coisas (objetos) para alcançar objetivos comuns [Atzori et al., 2010].

Tais elementos tecnológicos, nos quais informações podem ser recolhidas, processadas e acessadas a partir de “coisas”, permitem a modelagem computacional de sistemas independentes anteriormente, formando ambientes inteligentes que trocam dados e informações. Por exemplo, situações, atividades e processos que estavam presentes apenas no

domínio físico podem ser virtualizados, sendo utilizados na geração de relatórios, gráficos e animações que facilitem e agilizem o entendimento da situação de um projeto em tempo real. Ambientes inteligentes também ajudam na melhoria da automação de plantas industriais, usando de uma implantação massiva de etiquetas RFID associadas às peças de produção.

Com as expectativas crescentes criadas em torno do conceito de Internet das Coisas, e o papel que este conceito desempenhará em um futuro próximo, grandes empresas e consórcios, como o europeu financiado pela União Européia e formado por empresas como SAP e IBM, além de universidades como a de *Sapienza* de Roma e a *St. Gallen* da Suíça, vem trabalhando em definições de padrões de modelagem, projeto e desenvolvimento para a IoT. Alguns destes modelos são a EPCglobal [Huang and Li, 2010] e a arquitetura UID [Koshizuka and Sakamura, 2010]. Embora existam iniciativas a fim de criar um padrão para IoT tal como *UID Center* [uID Center, 2011], não há um padrão definido para este tipo de sistema.

O primeiro passo para o uso de IoT no *Framework* de Acompanhamento de Produto-Serviço (FAPS) é a aplicação do conceito em sistemas de PSS, incorporando componentes como *tags* RFID, sensores e outros nos recursos e insumos industriais passíveis de serem rastreados e acessados no ambiente. Assim, através de um conceito de abstração, que será definido futuramente no *framework*, cada entidade irá representar um elemento que tem uma marcação do tipo de componente. Com o auxílio de um *middleware*, que visa fornecer uma estrutura de comunicação entre entidades físicas, os dados de ambientes reais são coletados e utilizados na representação de ambientes virtuais. Com estes ambientes inteligentes, é possível a criação de sistemas gerenciais que mostram panoramas completos, detalhados e sem interação homem-máquina dos dados constantemente capturados e interpretados por componentes, como os de RFID, por exemplo. O *middleware* fornece a estrutura necessária para solucionar três aspectos fundamentais na implantação e operação de sistemas PSS: abstração (dados de aquisição e processamento), comunicação (entre os diferentes dispositivos) e representação (interface externa e resultados).

2.4 Arquitetura Orientada a Serviços (SOA)

Muitas empresas possuem uma quantidade grande de aplicações que utilizam diversas tecnologias e plataformas, o que pode tornar a integração de todos esses sistemas uma tarefa difícil. A globalização e o comércio eletrônico estão cada vez mais acelerando as novas demandas dos consumidores e reduzindo o ciclo de vida dos produtos [Endrei et al., 2004]. A arquitetura orientada a serviços possui uma abordagem para construção de sistemas distribuídos e busca atender este novo mercado ágil e flexível. SOA está disposto a resolver o *gap* existente entre negócio e TI. O SOA busca estruturar e abstrair regras e funções de negócios, mantendo a flexibilidade e heterogeneidade dos sistemas. SOA é um paradigma, ou seja, uma nova forma de pensar e desenvolver sistemas distribuídos [Josuttis, 2007].

Para [MacKenzie et al., 2006], SOA é definida como “um paradigma para organização e utilização de competências distribuídas que estão sob controle de diferentes domínios proprietários”.

A ideia do SOA para o modelo de referência desenvolvido pela [MacKenzie et al., 2006] é disponibilizar uma solução desenvolvida por uma entidade com o intuito de resolver um determinado problema de outras entidades, pois esta solução/competência pode ser compatível com as necessidades existentes. Desta forma, não existe uma relação de um para um entre as competências e as necessidades das entidades, fazendo com que a granularidade possa variar do fundamental ao complexo. Enquanto as competências e necessidades existem independentes do SOA, no ambiente SOA os serviços são formas de as unir. Em geral, as entidades que oferecem competências são vistas como provedores de serviços e as entidades que fazem uso destes serviços para atender as suas necessidades são denominadas como consumidores de serviços.

Portanto, para que seja possível a interação entre os serviços que implementam as competências e necessidades das entidades é necessário a existência de 3 serviços:

- Consumidor: entidade ou outro serviço que necessita de um serviço e o executa com auxílio de um middleware;
- Provedor: entidade que aceita e executa as requisições dos consumidores;
- De Registro: contém um repositório de serviço disponíveis e permite a procura des-

tes. A colaboração entre estes serviços segue o paradigma de “encontrar, conectar e invocar”, onde o consumidor realiza uma localização dinâmica do serviço requisitado que foi publicado anteriormente por um provedor, esta localização é realizada por meio de consulta no serviço de registro. Após encontrado, o consumidor conecta e invoca o serviço solicitado [Endrei et al., 2004].

Na arquitetura orientada a serviços, interoperabilidade e baixo acoplamento são conceitos importantes que devem ser vistos com mais detalhes [Josuttis, 2007].

- **Serviço:** SOA é focado em processos de negócio e estes processos são realizados por meio de atividades e tarefas em diferentes sistemas, assim, podemos dizer que um serviço corresponde a uma atividade ou tarefa de um processo de negócio. Tecnicamente, um serviço é uma interface para mensagens que retornam informação ou alteram o estado de uma entidade. De acordo com este conceito tudo que pode ser usado como uma interface e representa uma funcionalidade de negócio, pode ser chamado de serviço, mas nem toda interface pode ser considerada serviço em um ambiente SOA. Por exemplo, uma interface que permite criar, ler, alterar e excluir um usuário de sistema não é um serviço, quando as mesmas funcionalidades separadamente podem ser descritas em uma interface bem definida, representando o comportamento e a semântica de cada serviço disponibilizado. Em suma, serviço é a representação de alguma funcionalidade de negócio utilizando uma determinada tecnologia de informação, como por exemplo, “criar usuário” ou “informar parâmetro”.
- **Interoperabilidade:** Em ambientes heterogêneos, a principal meta a ser alcançada é facilitar a conexão entre os sistemas neste ambiente e esta facilidade é conhecida como alta interoperabilidade. Portanto, em SOA, alta interoperabilidade é o início e não o fim. Interoperabilidade é a base para se implementar as funcionalidades de negócios (serviços) em sistemas distribuídos.
- **Baixo Acoplamento:** Baixo acoplamento é um conceito e não uma ferramenta para a minimização das dependências entre os sistemas. Quando dependências são minimizadas, as modificações têm seus impactos minimizados, e os sistemas ainda funcionam mesmo quando partes do mesmo não estão operando. A minimização das

dependências contribui para a tolerância à falha e a flexibilidade, que é exatamente o que precisamos em um ambiente SOA.

2.5 Convergindo os conceitos

Existem duas principais diferenças entre a abordagem PSS e a abordagem CoPS. Em PSS, a empresa fornecedora de um serviço ou produto acaba se tornando parte integrante da empresa na qual presta serviço, uma vez que não apenas entrega um produto, mas sim executa uma função dentro da estrutura de trabalho do consumidor, muitas vezes deslocando uma equipe que atua dentro da outra empresa executando tarefas e captando *feedbacks* importantes ao processo. Já na abordagem CoPS, uma solução desenvolvida é entregue ao cliente, não fazendo com que a empresa entre para o ciclo de negócio ou que uma equipe comece a executar uma função dentro da lógica do cliente. Isso significa que quando uma empresa contrata um PSS, ela obtém um sistema que abrange o produto final, mas que junto dele carrega outros serviços que podem constantemente se alterar de acordo com a necessidade do cliente, deixando de possuir um produto, mas sim, alugando um sistema que executa a sua necessidade. Quando se contrata um CoPS, o cliente continua adquirindo um bem, só que mantém a propriedade e responsabilidade por ele. A empresa fornecedora realiza a venda e manutenção, mas a propriedade pertence ao cliente.

Além destes dois conceitos, o FAPS utiliza também a IoT que fornece o ambiente necessário para que os diferentes componentes interajam e se comuniquem. É através desta camada de comunicação que a solução se torna integrada, criando uma rede de interações dos diferentes serviços executados em um projeto. Esta camada faz com que componentes de um sistema, como uma pessoa e um dispositivo RFID, que sozinhos não conseguem se comunicar, se tornem capazes de colaborar entre si. Isto amplia o conhecimento e enriquece a rede como um todo, tornando o conhecimento dentro do sistema muito mais amplo.

Já o SOA, mostra como devem ser encarados os diferentes componentes da rede que forma este sistema. Cada função desempenhada no sistema é um serviço. Existem vários tipos de serviços e vários componentes que podem desempenhar serviços diferentes e fornecer as informações que um dispositivo necessita. Essa abstração, assim como todo o

funcionamento do *framework* FAPS, será detalhada no Capítulo 3.

Pode haver um alto ganho de sinergia entre os conceitos apresentados neste capítulo. Uma abordagem que consiga os unir pode usar os pontos fortes de um para cobrir as fraquezas do outro, fortalecendo uma estrutura e criando um *framework* capaz de gerenciar o desenvolvimento de soluções completas para o mercado.

Assim, após introduzir os conceitos aplicados no Framework proposto neste trabalho, a Tabela 2.2 mostra as principais características aplicadas de cada conceito.

PSS	<ul style="list-style-type: none"> • Constante atualização das etapas de projeto, aplicação e manutenção através do feedback rápido de dentro da empresa. • Solução que não abrange apenas uma etapa do processo, mas todo o ciclo de vida, agregando valor ao sistema integrado.
CoPS	<ul style="list-style-type: none"> • Aplicado a bens de alto custo, com elementos de diferente natureza interconectados. • Capacidade de gerar pacotes de complexidade e abrangência diferente para cada tipo de cliente.
IoT	<ul style="list-style-type: none"> • Fornece uma camada de comunicação para dispositivos de natureza diferentes, interligando toda a rede de componentes.
SOA	<ul style="list-style-type: none"> • Minimiza problemas existentes em CoPS em relação a alterações durante o processo. • Fornece uma abstração de serviços necessária na obtenção de componentes e funcionalidades independentes uns dos outros.

Tabela 2.2: Características importantes de cada conceito utilizado no *framework*

Capítulo 3

Framework de Acompanhamento Produto-Serviço

Com os conceitos apresentados até o momento, este trabalho propõe um *framework* que busca unir as principais características e vantagens em uma solução integrada que forneça a estrutura necessária para gerenciar o fluxo de informações presente no desenvolvimento de projetos. O *framework*, chamado de *Framework* de Acompanhamento Produto-Serviço (FAPS), busca identificar em um processo de produção:

- Componentes que desempenham papel atuante;
- Tipos de informações que são trocadas ou que são pertinentes no processo;
- Diferentes setores ou áreas existentes;
- Serviços desempenhados por cada componente do sistema.

Inicialmente, se faz necessário entender quais são as partes envolvidas no sistema geral. Parte do funcionamento do *framework* depende da abstração realizada para identificar quem integra o processo e quem visualiza o que acontece de fora.

Para cada projeto, é necessário que o usuário tenha bem definido o Processo de Negócio, que descreve uma série de atividades ordenadas, desempenhadas em etapas, para atingir um objetivo. Este processo descreve o fluxo e transformação do material, informação, operações e decisões. Processos de negócio e suas atividades são estruturados e tem um co-

meço e um fim, assim como definidas suas entradas e saídas [Hammer and Champy, 2003] [Rummler and Brache, 1995] [SOA4All, 2010].

Automatizar e generalizar processos que possuem informações de diferentes naturezas, inúmeras variáveis e diferenças espaço-temporais é o grande desafio do FAPS. A automação de processos não significa necessariamente que todo o processo é executado automaticamente. Devido ao trabalho com soluções personalizadas de inúmeras características únicas e lógicas de troca de informação variadas, o FAPS deve fornecer uma estrutura onde o principal é disponibilizado para o desenvolvimento de uma solução. A estrutura de comunicação, abstração de componentes, e representações gráficas amigáveis ao usuário é fornecida, porém parte da solução deve ser construída de acordo com o foco de aplicação da solução. Baseado em algumas soluções de software existentes [SOA4All, 2010], o framework complementa este mecanismo com uma camada de serviço separada, a fim de os processos mais versáteis [Goetz, 2009]. O capítulo 4 mostrará o processo de personalização da solução, onde o foco é a Indústria Naval e *Offshore*.

Devido a variação de componentes, funcionalidades e requisitos de funcionamento no processo, se fez necessária a abstração dos integrantes do sistema, que na fase de projeto do framework foram definidos e separados em camadas constituintes da arquitetura detalhada a seguir.

O framework deverá integrar informações presentes ao longo de todo o ciclo de vida do produto, disponibilizando os diferentes serviços existentes. A arquitetura envolverá sistemas de monitoramento, inspeção e acompanhamento até o controle de máquinas e dispositivos, passando pela representação dos serviços, suas informações e processos, bem seu interfaceamento com o usuário.

De forma a implementar tais funcionalidades, este trabalho faz uso do conceito de hiperambiente [Carvalho et al., 2011]. Este se caracteriza por ser um “*locus*” onde informações de diferentes naturezas são agregadas, constituindo cenários de interação compostos por entidades físicas reais e virtuais.

FAPS - Framework de Acompanhamento Produto-Serviço

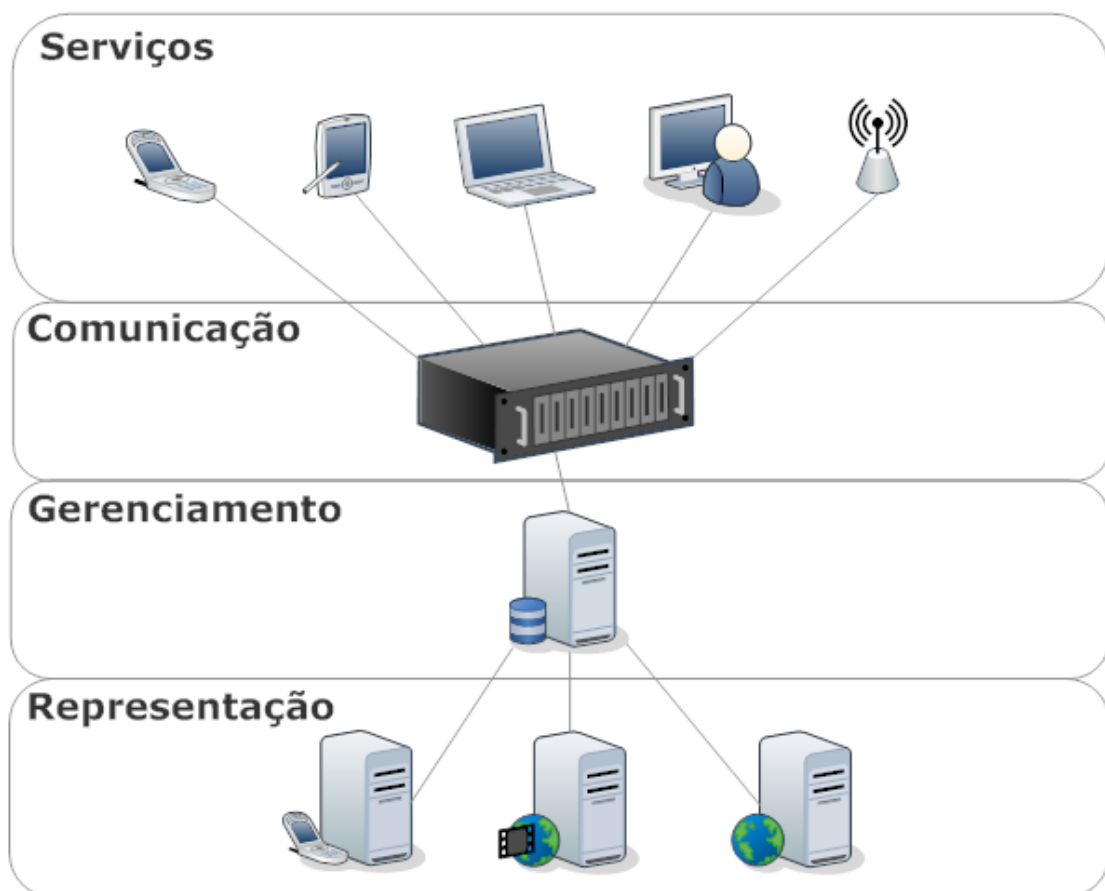


Figura 3.1: Representação do *Framework* de Acompanhamento Produto-Serviço em Camadas

3.1 O conceito de hiperambientes

Neste novo cenário tecnológico, novas teorias, metodologias, técnicas e ferramentas surgem, provenientes de diferentes áreas relacionadas à interface homem-computador. Com base no paradigma ubíquo e na IoT 3.4, definem-se novos modelos para a interligação de qualquer “coisa” (objetos, computadores, pessoas, etc) em uma rede, semelhante aos dispositivos já hoje interligados através da Internet.

A literatura carece de estudos que explicitamente considerem a combinação de vários ambientes inteligentes, com elementos naturais ou tecnológicos, incluindo a combinação de ambientes reais e virtuais. Embora existam estudos nesse sentido [Lifton and Paradiso, 2009], pode-se afirmar que nenhuma arquitetura ou ferramenta, que pode ajudar os desenvolvedores de aplicações, mecanismos e diretrizes criar este tipo de ambientes inteligentes, neste projeto chamados de hiperambientes, foi encontrada na literatura.

O conceito de hiperambiente será implementado através de um conjunto de técnicas e soluções computacionais, os quais compõem a arquitetura do FAPS.

3.2 Uma visão geral da arquitetura

A arquitetura deverá envolver cenários virtuais e reais, permitindo o cruzamento entre os dois mundos. Neste contexto, pode-se citar as obras “dupla realidade” [Lifton, 2007] e “*cross*-realidade” [Paradiso and Landay, 2009], que constitui ambientes resultantes da interligação entre os mundos real e virtual, através da mediação de redes de sensores e atuadores. Embora ambos sejam auto-suficientes, eles se beneficiam da capacidade de refletir, influenciar e mesclar uns aos outros.

A arquitetura apoia a fusão entre os vários elementos tecnológicos e sua representação virtual, dando aos designers e desenvolvedores o apoio necessário para a criação de hiperambientes que representarão o ciclo de vida de um produto.

A figura 3.2 apresenta a arquitetura proposta. Esta é composta por quatro camadas principais:

- Camada de serviços: congrega os principais elementos do sistema. É responsável

pela aquisição e tratamento das informações, os quais são encapsulados em serviços disponibilizados a outros componentes integrantes do FAPS. Cada elemento desta camada, seja um dispositivo de *hardware* ou um sistema manipulado por pessoas, tem suas informações abstraídas em serviços, facilitando o escalonamento e agrupamento de funções no sistema.

- Camada de comunicação: *middleware* responsável pela troca de informação entre os componentes fornecedores de serviços e o sistema gerenciador do FAPS. Tem a capacidade de fazer com que todo e qualquer elemento integrante do sistema consiga adquirir e fornecer informações em um formato universal dentro da arquitetura.
- Camada de gerenciamento: responsável pelo gerenciamento de todos os dados recebidos na rede de serviços do FAPS. Cada informação fornecida pelos dispositivos é armazenada, ao mesmo tempo também guardando histórico de requisições de dados e trocas de informações em um banco de dados. O sistema permite que aplicações da camada de representação tenham acesso aos dados necessários para a sua execução, possibilitando também execuções em tempo real ou em estágios anteriores de um certo processo.
- Camada de representação: conjunto de aplicações que com os dados provenientes dos diversos serviços presentes no ambiente podem executar alguma função ou fornecer alguma interface de interação ou melhor entendimento dos processos em andamento. Estas aplicações de representação podem ser interfaces *web*, animações 3D em tempo real ou mesmo simuladores físicos.

A seguir, detalha-se o projeto e implementação de cada uma das camadas.

3.3 Camada de Serviços

Um projeto industrial possui inúmeras entidades executando diversas tarefas simultaneamente. Muitas vezes estas não tem relação clara entre si, mas em um espoco geral do processo, informações podem ser relevantes no funcionamento de outros componentes. Claramente, existem muitas dificuldades que atrapalham o desenvolvimento de um sistema genérico capaz de unir atividades distintas ocorrendo nos processos. Pela grande

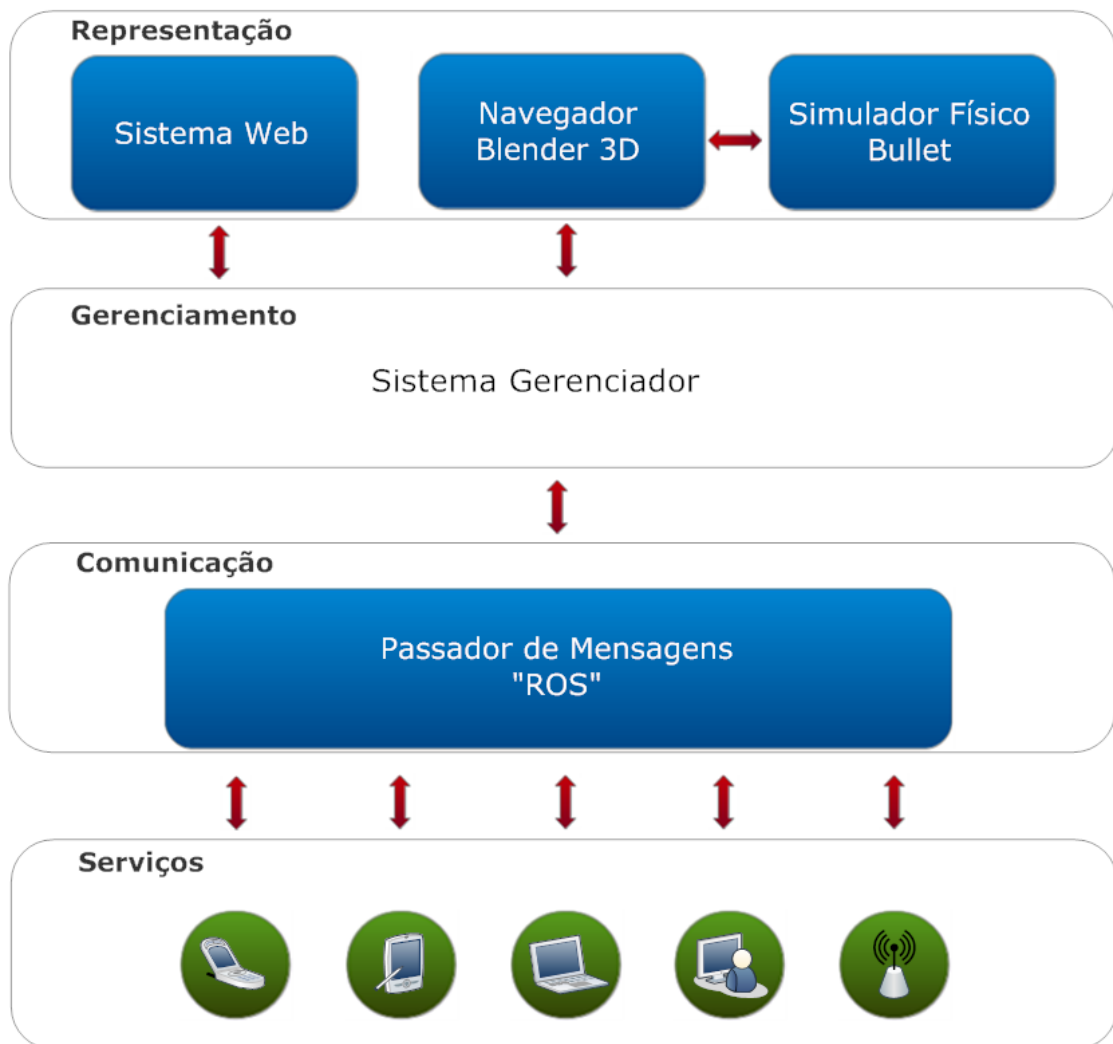


Figura 3.2: Arquitetura final do FAPS.

quantidade de dispositivos de natureza diferente, não há padronização nas informações que estes componentes fornecem.

Para uma solução integrada funcionar propriamente em um ambiente, o sistema deve ser capaz de receber entradas de dados de diversos atores do processo e interligar as ações e interações entre componentes. Componentes que desempenham papel importante no sistema podem ser máquinas, computadores, smartphones, robôs, sensores, leitores de radiofrequência, seres humanos e muitos outros, capazes de trocar informações entre si ou armazenar dados em um sistema gerenciador. Devido as diferentes entradas de dados proporcionadas por estas entidades, se faz necessária uma maneira de abstrair as informações fornecidas e armazená-las, facilitando protocolos de comunicação entre os dispositivos e componentes do sistema.

Tendo em vista a heterogeneidade da rede de dispositivos, foram definidas formas genéricas de armazenamento dos dados e informações. Cada componente do FAPS é definido pelos serviços que pode fornecer no sistema. Um GPS instalado em um caminhão, por exemplo, não é visto como um componente de *hardware*, mas sim como um serviço de localização. Qualquer componente que precise saber a localização do caminhão requisitará um serviço de localização. Na existência deste GPS, ele se encarregará de fornecer a informação necessária, não significando que apenas ele pode fornecer este serviço. Na situação de o GPS não estar presente ou apresentar problemas técnicos, um sistema auxiliar de logística, que prevê a localização aproximada do referido caminhão no momento da requisição, pode substituí-lo, pois este também oferece o serviço de localização requisitado.

Esta camada não é formada apenas por dispositivos de *hardware*, mas também por dados passados com interação humana ou de outros sistemas. Ela também se constitui por sistemas que já funcionam em uma empresa ou então que possuem informações e lógicas específicas para outros setores pertencentes ao sistema integrado que forma o FAPS, fornecendo um determinado serviço que pode ser de uso para outros componentes dentro da estrutura, ou mesmo algum serviço que alimente o sistema gerenciador do FAPS. Além de sistemas, as interfaces em dispositivos móveis, ou mesmo em computadores tradicionais, podem funcionar como meio para que ocorram entradas no sistema, sejam em uma aplicação local ou na web.

Além da abstração que trata todo componente como um serviço, é necessário também

definir padrões de comunicação entre os componentes presentes no sistema integrado. Devido aos diferentes “idiomas” falados pelos inúmeros dispositivos que podem existir em uma rede heterogênea, caso que pode facilmente acontecer na aplicação deste *framework*, é necessária a existência de um agente que faça a “tradução” das informações, tanto para entradas do sistema, quanto para saídas. Esta camada não possui os *drivers* responsáveis pela comunicação entre os dispositivos e o sistema operacional, por isso, não consegue se comunicar com o sistema gerenciador sem o auxílio de um *middleware*. Este *middleware*, camada responsável por esta comunicação, é quem consegue formatar as informações passadas pelo dispositivos para o formato reconhecido pelo sistema gerenciador. Depois de formatada, o *middleware* envia a informação para o sistema, processo que será melhor explicado na Sessão 3.4. Um banco de dados, estruturado para armazenar os dados do sistema, trata as informações recebidas e permite com que estas sejam utilizadas para controle das atividades do projeto, sendo mais bem detalhado na Sessão 3.5.

A partir destas informações são possíveis tomadas de decisões, geração de relatórios, alimentação de subsistemas e aquisição de entradas para atividades do processo.

3.4 Camada de Comunicação

Devido aos diferentes tipos de dispositivos pertencentes à rede de componentes de um FAPS, são necessárias maneiras de se lidar com os diversos tipos de informação existentes no ambiente. Ambientes inteligentes, como no caso do que é proposto neste trabalho, são sistemas distribuídos e heterogêneos, que podem incluir serviços de apresentação multimídia, componentes de automação e controle de produção, objetos físicos inteligentes, sensores *wireless*, dispositivos móveis pessoais ou compartilhados, e muitos outros sistemas e entidades. Essa complexa configuração requer suporte em diversas tarefas, entre elas: visualização do estado do sistema, simulação e acompanhamento dos componentes em um ambiente, juntamente com suas propriedades físicas e digitais. Um pré-requisito para este suporte é um *middleware* que permita a conexão de todos estes componentes heterogêneos distribuídos, tanto diretamente, por códigos-fonte abertos e por *drivers* proprietários, quanto por troca de pacotes, por exemplo. *Middlewares* são tecnologias para o desenvolvimento, implantação, execução e interação de aplicações. Estas camadas de

software estão posicionadas entre os sistemas operacionais e as aplicações [Ibrahim, 2009]. Este *middleware* se conecta a um servidor mestre e a nodos descentralizados interligados, através de diferentes tipos e topologias de rede [Roalter et al., 2011]. Um dispositivo integrante do *framework* tem de ser capaz de passar um dado para outro componente da mesma rede. Cabe ao componente que recebe a informação interpretá-la e concluir se essa é importante ou não.

Inicialmente, pensou-se em utilizar o formato XML para armazenar as mensagens que precisam ser transferidas de um componente para outro, assim como para o sistema, por sua simplicidade e maleabilidade de estrutura, podendo ser facilmente escrito e lido por diversas aplicações. O XML tem sua estrutura desenvolvida diretamente para o uso na internet, com suporte amplo de linguagens, facilidade de uso e manipulação rápida [W3C, 2008]. Infelizmente, apesar da facilidade de uso do formato, dispositivos industriais, como sensores, leitores e máquinas, tradicionalmente não são dispositivos web e não possuem uma interface de comunicação *web* por *default*. Buscou-se então uma alternativa que conseguisse facilitar a comunicação de dispositivos que transmitem informações em formatos diferentes, fazendo com que esses possam compreender os dados vindos de outros componentes.

A alternativa encontrada foi utilizar o ROS (*Robot Operating System*), sistema operacional que consegue comunicar diversas naturezas de dispositivos eletrônicos. Apesar de amplamente utilizado para sistemas robóticos, o ROS possui funcionalidades de personalização para diferentes tipos de demandas e integrações, facilitando a utilização com sistemas e *hardware* customizado. Formalmente, o ROS é definido como um sistema de código-fonte aberto e meta-operacional para robôs, que fornece serviços esperados de um sistema operacional, incluindo abstração de *hardware*, controle de dispositivos de baixo-nível, implementação de funcionalidades comumente usadas, passagem de mensagens entre processos e gerenciamento de pacotes. Ele também fornece ferramentas e bibliotecas para a obtenção, construção, escrita e execução de código em vários computadores [ROS, 2012]. O ROS foi originalmente desenvolvido pelo Laboratório de Inteligência Artificial da Universidade de *Stanford* e atualmente é uma plataforma livre, lançado sob a licença BSD, para o desenvolvimento de redes inteligentes de sensores e atuadores, que vão desde aplicações para ambientes de realidade aumentada a computação ubíqua e redes de sensores.

Com a utilização generalizada no meio acadêmico e comercial, o ROS está sendo considerado a principal ferramenta para desenvolvimento de aplicações de robótica, fornecendo bibliotecas e aplicações para auxiliar no desenvolvimento de sistemas autônomos de robótica e personalizados. O ROS está disponível para sistemas operacionais do tipo UNIX como MacOS e Linux, sendo recentemente disponibilizado também para o sistema operacional Android. Assim, o ROS aumenta o seu potencial de uso, permitindo novos recursos e aplicações.

De acordo com [Roalter et al., 2011], o ROS proporciona a utilização de uma série de vantagens: i. Redução da quantidade de tempo gasto para implementar métodos de comunicação abstrata e básica; ii. evitar a adaptação do *middleware* de um problema específico para nossa pesquisa; iii. Lucro de uma grande comunidade melhorando *drivers* e *middleware*; iv. Compatibilidade do *middleware* com um grande conjunto de sistemas de *hardware*. Outras características do uso de ROS como *middleware* pode ser encontrada em [Kranz et al., 2011].

A base estrutural do ROS fornece um servidor chamado aqui de *roscore*, responsável por permitir que diferentes componentes se localizem e troquem informações. Já os objetos de interesse, como sensores, *grid* de sensores, atuadores, objetos inteligentes, etc. são chamados de *rosnodes* neste trabalho. No ROS, existem duas maneiras de *rosnodes* se comunicarem:

- se inscrevendo em um *rostopic*, que funciona como um canal de divulgação de dados, onde todos os inscritos recebem dados dos publicadores do tópico;
- através de *rosmessages*, que são mensagens que podem ser criadas e personalizadas da maneira desejada para melhor comunicar componentes.

A principal contribuição do *framework* FAPS, é criar um servidor de mensagens acoplado ao seu funcionamento, onde qualquer informação que necessite ser transmitida de um componente a outro será enviada através dessa estrutura. Esta fica responsável em informar o componente de destino desejado, abstraindo questões de compatibilidade ou heterogeneidade de *hardware*. A camada de comunicação pode ser vista como o meio de passagem de mensagens entre os componentes do FAPS. Quando um componente necessita de um dado de outro componente ou do banco de dados do sistema gerenciador, a

requisição e seus respectivos parâmetros são formatados em uma mensagem ao sistema gerenciador, que passa a requisição para o componente de destino. Após processar a requisição o componente retorna a informação ao sistema gerenciador, que repassa ao componente interessado. Assim, se cria o cenário de abstração citado no capítulo 2, onde todos os componentes do *framework* são vistos como fornecedores de serviços, que juntos formam uma solução de produto-serviço - PSS.

3.5 Camada de Gerenciamento

O sistema gerenciador é a parte central do FAPS. Ele possui acesso a todas as informações que já foram captadas por componentes, fazendo a interação de todos os integrantes da rede com o banco de dados, e principalmente, comanda o funcionamento das atividades de cada projeto. O sistema pode ser visto como um grande repositório de dados, onde estarão armazenadas todas as informações transmitidas pelos dispositivos e sistemas existentes da camada de serviços. O banco de dados, que utiliza o Sistema de Gerenciamento de Banco de Dados (SGBD) do tipo MySQL, possui tabelas que armazenam informações dos componentes cadastrados no sistema. Estas informações podem ser:

- Dados de identificação e informações como nome, tipo de componente e status atual no sistema;
- Tipos de informação que o componente pode fornecer, assim como se estas estão disponíveis ou não;
- Histórico de informação de componentes;
- Dependências em relação a outros componentes no sistema.

A principal função desta camada é permitir que sistemas que dependem dos dados fornecidos pelos componentes do FAPS sejam capazes de adquiri-los de maneira simples e padronizada. Sistemas de controle administrativo, de WBS (*Work Breakdown Structure*) ou mesmo interfaces 3D podem utilizar os dados vindos do sistema gerenciador para alimentar o seu funcionamento. Com o sistema gerenciador, é possível manter um histórico de todas as etapas que ocorreram ao longo dos processos até um certo ponto. Nenhuma

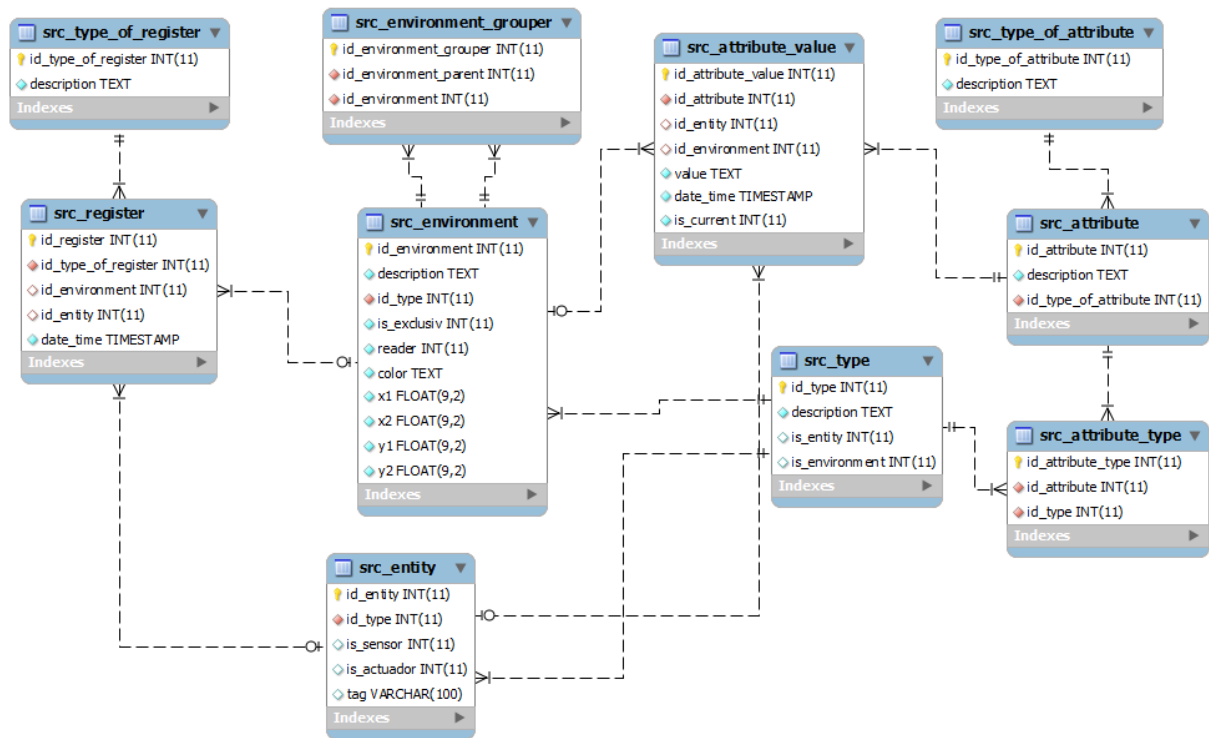


Figura 3.3: Estrutura de tabelas do banco de dados MySQL.

informação é perdida, permitindo análises ao longo do tempo e estatísticas que podem auxiliar em diversas tomadas de decisão.

Para o desenvolvimento deste *framework*, manter a forma de acesso às informações a mais genérica possível foi foco constante em cada etapa, visto que, como comentado anteriormente na sessão 2.4, a heterogeneidade da rede é muito grande. Assim, com a ajuda do ROS, explicado na sessão 3.4, foi possível solucionar os problemas oriundos de diferenças entre componentes participantes na rede. O sistema gerenciador nada mais é que nodo conectado ao servidor roscore. A diferença é que, diferente de outros componentes, o sistema gerenciador se inscreve em todos os tópicos de comunicação, monitorando constantemente tudo que acontece na rede e armazenando as informações para uso futuro. Além disso, ele cria um tópico próprio, onde recebe e publica dados que não estão presentes em tempo real, mas sim precisam ser calculados ou consultados em estados anteriores do sistema. Os dados referentes a cada componente são armazenados no banco de dados e ficam disponíveis para qualquer consulta futura.

Na estrutura de tabelas do BD, mostrada na figura 3.3, é previsto que qualquer componente pode ter inúmeros atributos com quaisquer formas de valores. Isso é possível graças a disposição das tabelas do banco. Para uma tabela representar os atributos e informações que todos os componentes presentes na rede do FAPS possam vir a transmitir, atributos estáticos na tabela se tornam impraticáveis, visto que a cada novo tipo de informação, alterações deverão ser realizadas na estrutura do banco de dados. Assim, a solução encontrada foi permitir que atributos, seus tipos de informação e seus valores sejam adicionados dinamicamente, em tabelas separadas ligadas entre si. Na figura 3.3, as tabelas “src_entity” (entidade), “src_attribute” (atributo), “src_attribute_value” (valor do atributo), “src_type_of_attribute” (tipo do atributo). Como exemplo, pode-se analisar um dispositivo de hardware que fornece constantemente uma informação de temperatura para o sistema. Ele possuirá uma entrada para o dispositivo na tabela de entidades, e seu valor de leitura será armazenado na tabela de valores, onde será armazenada a informação referente a qual atributo aquele valor pertence (temperatura) e também qual é o tipo de dado do atributo (Celsius, por exemplo). Se em algum momento o dispositivo necessitar o armazenamento de outro atributo, como medição de pressão, por exemplo, ou mesmo trocar o tipo de leitura para Kelvin, nenhuma alteração estrutural do BD será necessária.

3.6 Camada de Representação

Uma vez que todas as informações do sistema estão armazenadas em um banco de dados, é difícil numerar a quantidade de possibilidades em relação ao uso e modos de representação destes dados. Com os dados brutos, é possível moldá-los para inúmeros tipos de representação. A camada de representação busca minerar estas informações presentes no banco de dados e usá-las de forma a passar a maior quantidade de conhecimento possível ao usuário final.

A primeira parte integrante da camada de representação é a interface *web*, demonstrada nas figuras 3.4 e 3.5. Foi desenvolvida uma plataforma de acesso a todas as informações armazenadas no banco de dados, facilitando ao usuário final o acesso e a visualização das informações cadastradas. A figura 3.4 mostra a tela inicial do sistema *web*, que centraliza o acesso a todos os dados armazenados no sistema, ilustrando um exemplo

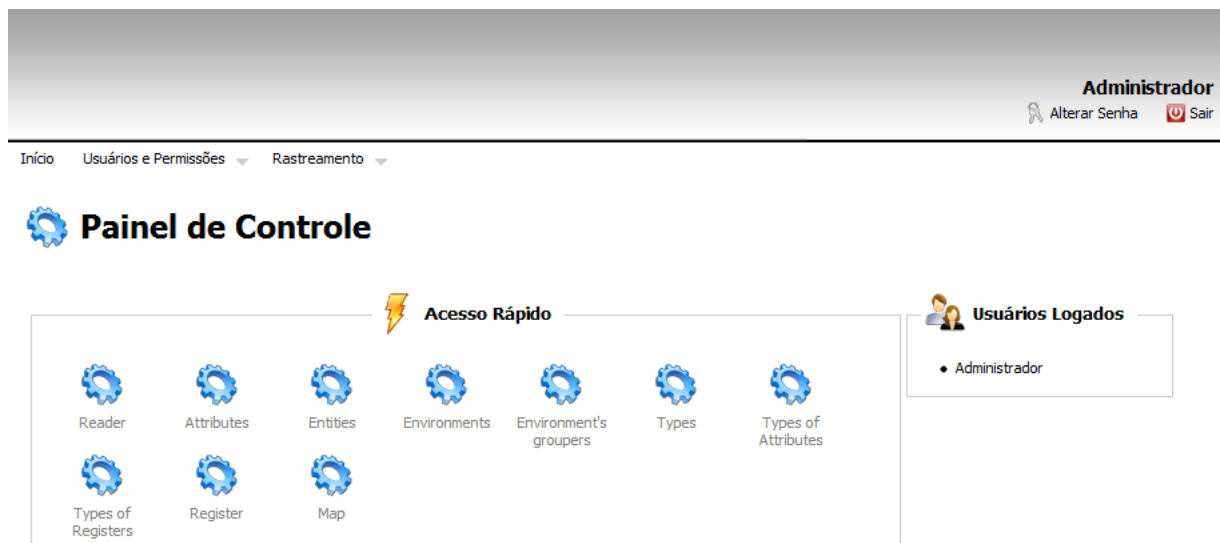


Figura 3.4: Página inicial do sistema *web* de representação dos dados armazenados no banco de dados do FAPS.

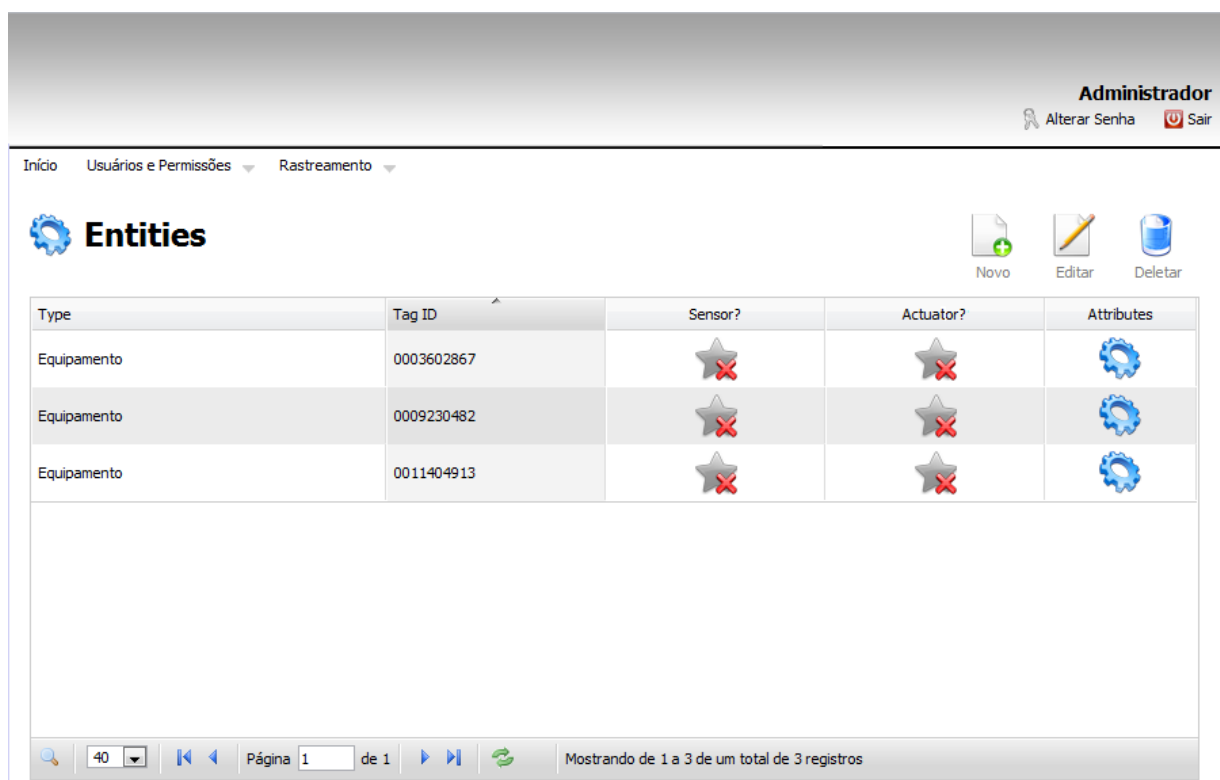


Figura 3.5: Exemplo de tela de listagem no sistema *web* de representação dos dados armazenados no FAPS.

de sistema que comunica com o banco de dados do FAPS. Já a figura 3.4 demonstra a listagem de entidades cadastradas no sistema. Nesta tela é possível também a inserção de novos valores e novos integrantes do sistema, fazendo com que a interface *web* não seja apenas uma maneira de acessar os dados, mas também modificar ou acrescentar qualquer informação desejada. A aplicação foi desenvolvida utilizando: a linguagem de programação PHP, por seu objetivo de permitir que desenvolvedores *web* escrevam páginas dinâmicas geradas rapidamente [PHP-Group, 2001]; a biblioteca JQuery, que simplifica a manipulação de eventos, animações e interações [jQuery Foundation, 2012]; conceitos de padrão MVC (*Model-View-Controller*), onde a separação em componentes de modelo, visão e controle permite independência para desenvolvimento, teste e manutenção de cada componente [Pham, 2010]. A estrutura da aplicação permite que os dados presentes no banco de dados compartilhado pelo FAPS possam ser modelados e utilizados para outras funcionalidades.

Através de relatórios, por exemplo, o usuário se torna capaz de visualizar dados de maneira direta e focada, alcançando rapidamente o resultado desejado. A camada de representação, manipulando os dados armazenados no banco de dados, pode adquirir informações como o número de horas de trabalho de um funcionário, número de horas em que um equipamento foi utilizado, tempo ocioso de estações de trabalho, quais atividades estão atrasadas ou demorando tempo além do previsto e muitas outras. Muitas vezes estes dados precisam ser trabalhados para se tornarem visíveis ao usuário. Qualquer gerente ou coordenador de equipes em uma empresa precisa de ferramentas que agilizem e auxiliem o gerenciamento das atividades, facilitando o rápido entendimento das situações e apresentando indicadores de desempenho capazes de demonstrar ao usuário o cenário de forma confiável. Indicadores de Desempenho (*Key Performance Indicators - KPI*) são definidos como metas de valores para certa tarefa ou processo dentro de uma empresa, como, por exemplo, cumprir certo processo em menos de três dias. KPIs são influenciados por métricas baseados em dados de tempo de execução do processo e medem o sucesso do processo como um todo [Wetzstein et al., 2009].

Além de relatórios, os dados existentes no banco de dados do sistema também podem ser utilizados para representar, em modelos 3D, os acontecimentos ocorridos ao longo de todo o ciclo de vida do produto. O FAPS apresenta em sua camada de representação um

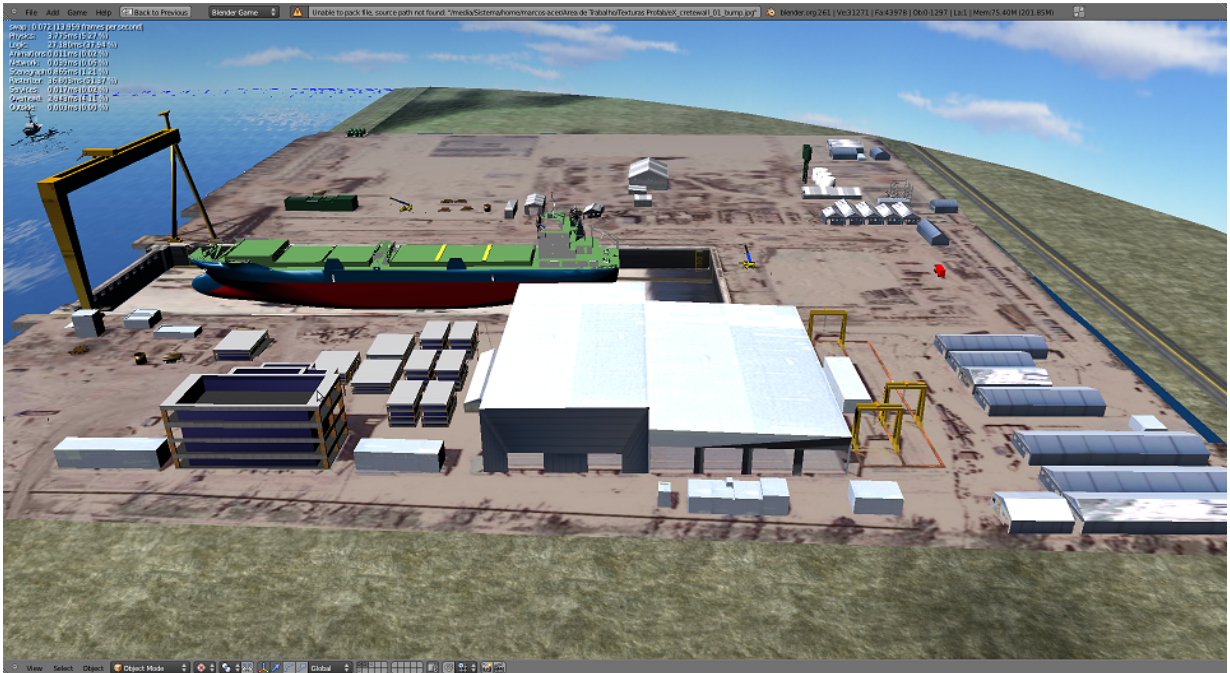


Figura 3.6: Maquete virtual de um estaleiro.

módulo de visualização avançado desenvolvido a partir da ferramenta Blender (figura 3.6), que permite descrever componentes dentro de maquetes virtuais através de *scripts* Python, os quais recebem as informações da camada de gerenciamento. Blender é um pacote de ferramentas que permite a criação e reprodução linear e em tempo real de conteúdo 3D interativo. Ele oferece a funcionalidade completa para modelagem, renderização, criação de animação e outras funcionalidades [Blender, 2003]. A linguagem Python, por sua vez, como outras linguagens de *script*, tende a facilitar o rápido desenvolvimento, onde não é necessária preocupação com problemas de memória e outras questões presentes em linguagens como C e C++ [Mein, 2004].

Com base na arquitetura apresentada, o FAPS teve sua implementação realizada. O *framework* foi testado e validado em uma aplicação associada à indústria naval, apresentado no capítulo seguinte.

Capítulo 4

Aplicando o FAPS na Indústria Naval

Com base no *framework* descrito anteriormente, este capítulo apresenta um estudo de aplicação do FAPS a um segmento industrial de construção e montagem de grandes estruturas. Foi escolhido realizar esta análise junto ao ciclo de vida de um navio. Isso se deve a sua complexidade, as expectativas de graus de competitividade internacional do segmento e, finalmente, a presença de um polo naval e *offshore* em instalação na região onde este trabalho foi realizado.

A seguir apresenta-se o panorama da construção naval brasileira de forma a subsidiar a aplicação e validação do FAPS.

4.1 A Indústria Naval Brasileira

No final da década de 1990, a indústria brasileira de construção naval enfrentou um período de queda vertiginosa dos níveis de produção, que a literatura associa a fatores como [Lima and Velasco, 1998] [Passos, 2007] [Favarin, 2008]: i. Crise do petróleo, em nível mundial; ii. Concessão indiscriminada de subsídios por um longo período de tempo (mais de 20 anos), sem exigências de aumento de produtividade que obrigasse o aumento da competitividade internacional da indústria; iii. Dependência de encomendas do setor estatal (Petrobrás, Vale e Lloyd Brasileiro); iv. Longo período de instabilidade econômica e inflação elevada, o que afetou toda a indústria de bens de capital sob encomenda e, em especial, a construção naval, que demanda dois anos, em média, por obra e administra centenas de fornecedores; v. Deficiências gerenciais nos segmentos financeiro e administrativo

dos estaleiros; vi. Não confiabilidade em relação ao cumprimento dos prazos contratuais de entrega (destaca-se que dos 61 navios financiados no período 1985/94, apenas 15 foram entregues rigorosamente dentro do prazo contratual).

O início da revitalização da indústria de construção naval no Brasil se dá, prioritariamente, a partir do final dos anos 90, com a criação de leis, por parte do governo, que flexibilizaram a exploração e produção do petróleo brasileiro. Associado a este fator, o desenvolvimento de novas tecnologias de exploração de lâminas d'água ultra-profundas pela Petrobras demandou contratação dos serviços de embarcações de apoio marítimo no início dos anos 2000, via licitações, as quais originaram encomendas aos estaleiros nacionais [Araujo et al., 2010]. Com o crescimento da demanda e com o incentivo ao setor naval nacional, medidas de inovação que aumentem a competitividade, além de outros fatores, se tornam essenciais para alcançar melhores resultados. A computação e suas áreas de PDI podem desempenhar um papel muito importante nesta tentativa.

A ideia de usufruir da computação e da automação nas principais atividades dentro dos estaleiros já é difundida e bem documentada, com exemplos nos principais estaleiros do mundo. Por exemplo, segundo [Jia and Jinke, 2011], o principal problema que a construção naval chinesa enfrenta é a baixa produtividade, e a informatização é um caminho para resolver este problema. Portanto, promover a informatização da indústria de construção naval tem uma grande importância para reforçar a competitividade global.

O setor naval brasileiro possui potenciais aplicações para a tecnologia da informação em seus processos, mas que ainda não foram completamente utilizadas e interligadas. Atualmente, o cenário é de controle através de planilhas *Excel* ou sistemas que não se comunicam, causando trabalho de preenchimento e de importação para que as informações se mantenham consistentes nas diversas bases de dados existentes. No estudo de [Jia and Jinke, 2011], a necessidade de unificar as bases de dados e os sistemas foi percebida, com o propósito de permitir a universalidade da informação.

Os Sistemas Complexos de Produção possuem a característica de incluir o cliente diretamente no seu funcionamento, integrando suas necessidades e provendo uma solução que seja totalmente focada na personalização e na melhoria dos processos. O setor naval brasileiro fornece um cenário ideal para esta abordagem, visto a participação majoritária da Petrobras ao longo de todo o ciclo de vida do navio/plataforma encomendada. Desta

forma, abordagens de produção baseadas na identificação das diferentes etapas do seu ciclo de vida, de forma a integrar informações, re-usar e customizar procedimentos, presentes no paradigma CoPS/PSS, podem ser utilizadas no panorama atual da indústria naval brasileira.

Assim, será estudada a viabilidade e adequação da ferramenta FAPS a implantação de paradigmas CoPS/PSS ao longo do ciclo de vida de um navio. Com a identificação das etapas e processos/serviços existentes, o FAPS permitirá o monitoramento e histórico das atividades desempenhadas na parte de projeto, construção, uso e desmonte de embarcações, focando em auxiliar a visão de agências proprietárias e investidoras. Os setores operacionais, de procurement, administrativos e, até mesmo, de manutenção, desmonte e reciclagem de navios poderão ser beneficiar com o *framework* FAPS, através do acesso as diferentes informações, disponíveis através de serviços, ao longo do ciclo de vida do navio, de forma a retroalimentar cada uma de suas fases, fornecendo melhorias e otimizações de processos.

4.2 Apropriando Conceitos CoPS ao Ciclo de Vida de um Navio

A utilização dos conceitos de ES ao longo do ciclo de vida do navio permite a visão geral do estado dos desejados setores, equipamentos, componentes e processos trabalhados, presentes no ambiente de estaleiro, bem como quando o processo de edificação de um navio ou plataforma termine, mantendo-o e continuando sua atualização até o seu desmonte.

Grandes empresas de solução de TI, atualmente, realizam investimentos na área naval, visto o potencial de lucro e melhorias no setor. A IBM, por exemplo, devido a sua parceria com a Dassault Systèmes, empresa francesa desenvolvedora da plataforma CAD/CAM/CAE chamada CATIA® e o sistema de simulação DELMIA®, possui trabalhos de pesquisa e propostas de solução na área de Gerenciamento do Ciclo de Vida do Produto (PLM - Product Lifecycle Management), que abrangem o processo completo de construção naval para embarcações. As atividades compreendidas vão desde a engenharia e coordenação do projetos, até a manutenção e operação dos produtos, usando de recursos de computação gráfica e de planejamento [IBM, 2008].



Figura 4.1: Representação gráfica do Ciclo de Vida de um navio.

Devido a complexidade inerente em desenvolver um sistema que abrange todas as áreas do ciclo de vida de um navio, o FAPS busca fornecer um repositório de dados universal ao navio, onde toda informação recebida por um dispositivo componente do ciclo é armazenada para uso futuro por outra aplicação ou outro dispositivo.

Utilizando os conceitos apresentados até aqui, podemos analisar que nos estaleiros, funcionários, equipamentos e máquinas podem ser identificados como componentes que, juntos ou individualmente, fazem parte dos seus diversos ambientes. Cada componente é capaz de receber ou fornecer serviços e cada ambiente possui seus componentes individuais ou compartilhados, onde todos fazem parte do ciclo de vida do navio. Assim, com esta forma de abstração, é possível organizar e filtrar dados relevantes a cada setor, mesmo que fisicamente estes não se comuniquem.

Com base em conceitos CoPS/PSS, o FAPS trabalha com uma separação do ciclo de vida do navio que pode ser vista simplificada na figura 4.1. Serão identificados os possíveis serviços e informações componentes do ciclo. Os processos de Projeto e Construção de uma embarcação pertencem ao início do ciclo de vida do navio. Após surgem as fases de fretamento e uso da embarcação, finalizando com seu desmanche e reciclagem. A seguir, tais etapas e sua análise são descritas com maior detalhamento.

BoL - *Beginning-of-life* No BoL de uma embarcação, é possível usufruir do FAPS ao longo das suas principais etapas, descritas a seguir:

- Projeto: o FAPS será capaz de gerenciar as atividades de projeto, congregando as informações de CAD, escalonamento da produção, orçamentos e fornecedores. Estas informações serão disponibilizadas e atualizadas ao longo de todo o ciclo de vida, sinergindo e resultando em melhorias em seus próprios processos, assim como em processos de outros elementos. O controle por parte de chefes de divisão e gerentes também pode ser auxiliado, visto que a aquisição em tempo real de seu estado estará disponível a qualquer momento, seja atual ou de períodos passados.
- Construção e Montagem: diferentes componentes e serviços podem compor o FAPS no que tange a etapa de construção e montagem. Destes pode-se citar:
 - identificação, acompanhamento e rastreamento de chapas metálicas, do pátio até sua transformação;
 - identificação, acompanhamento e rastreamento de peças;
 - identificação, acompanhamento e rastreamento de blocos e estruturas;
 - identificação, acompanhamento e rastreamento de funcionários;
 - monitoramento de condições de SMS;
 - identificação, supervisão e rastreamento de insumos;
 - identificação, monitoramento e rastreamento de equipamentos nas oficinas e pátios;
 - monitoramento das condições de segurança e funcionamento de oficinas e câmaras de pintura;
 - visualização e simulação de operações de logística e içamento;

MoL - *Middle-of-life* : uma vez finalizada a sua construção o navio passa as etapas de fretamento e uso, constituindo o meio do seu ciclo de vida. Nestas etapas, identificam-se os seguintes componentes e serviços a serem incorporados pelo FAPS:

- Comissionamento: ao longo da construção do navio, estabelece-se o processo de comissionamento, o qual é intensificado na sua finalização. Durante o comissionamento, diversos serviços podem ser identificados e monitorados. Estes podem ter

representações 3D associadas ao seu projeto e localização na planta, bem como o estado e histórico destes, para serem consultados quando necessário; Serviços de ajuda aos procedimentos de teste e re-arme poderão estar disponíveis;

- **Manutenção:** os procedimentos de manutenção, sejam eles preventivos/preditivos ou não, também serão tratados como serviços dentro do FAPS, utilizando o mesmo *framework* PSS utilizado no BoL. As informações atualizadas do *framework* estarão disponíveis temporalmente vindo a realimentar inclusive o projeto inicial do navio, da mesma forma que novas decisões de projeto no BoL poderão impactar os serviços de manutenção. Serviços de alarmes e agendamentos também podem ser desenvolvidos com sistemas que utilizem a base de dados do FAPS.

EoL - *End-of-life* No final do ciclo, no fim da vida de um navio, quando este já tem muitos anos de serviço ou quando reparos e adaptações não são justificados financeiramente, o navio é reciclado. Normalmente, o armador do navio o vende a um ferro-velho de navios para a demolição. Ainda assim, o navio pode fornecer alguns serviços e produtos importantes para a realimentação do ciclo e a criação de um novo navio. Dados sobre a estrutura, material e acabamento podem ser utilizados em comparação a novos projetos, mas principalmente, chapas de aço uma vez identificadas no PSS, podem ser reaproveitadas, depois de passar por processos de recuperação, servindo de material reciclado que fará parte do projeto de uma nova embarcação, reiniciando o ciclo de vida de um navio.

4.2.1 Utilizando o FAPS no Ciclo de Vida de um Navio

Com base na identificação das etapas CoPS ao longo do ciclo de vida de um navio, apresenta-se a seguir um caso de uso do *framework* para a disponibilização de serviços decorrentes do BoL.

A arquitetura FAPS-Navio

Com base na arquitetura FAPS apresentada no capítulo 3, apresenta-se a seguir uma instanciação dessa, denominada FAPS-Navio, para o seu uso ao longo do ciclo de vida de uma navio. Como o estudo de caso deste trabalho será focado no BoL, a arquitetura

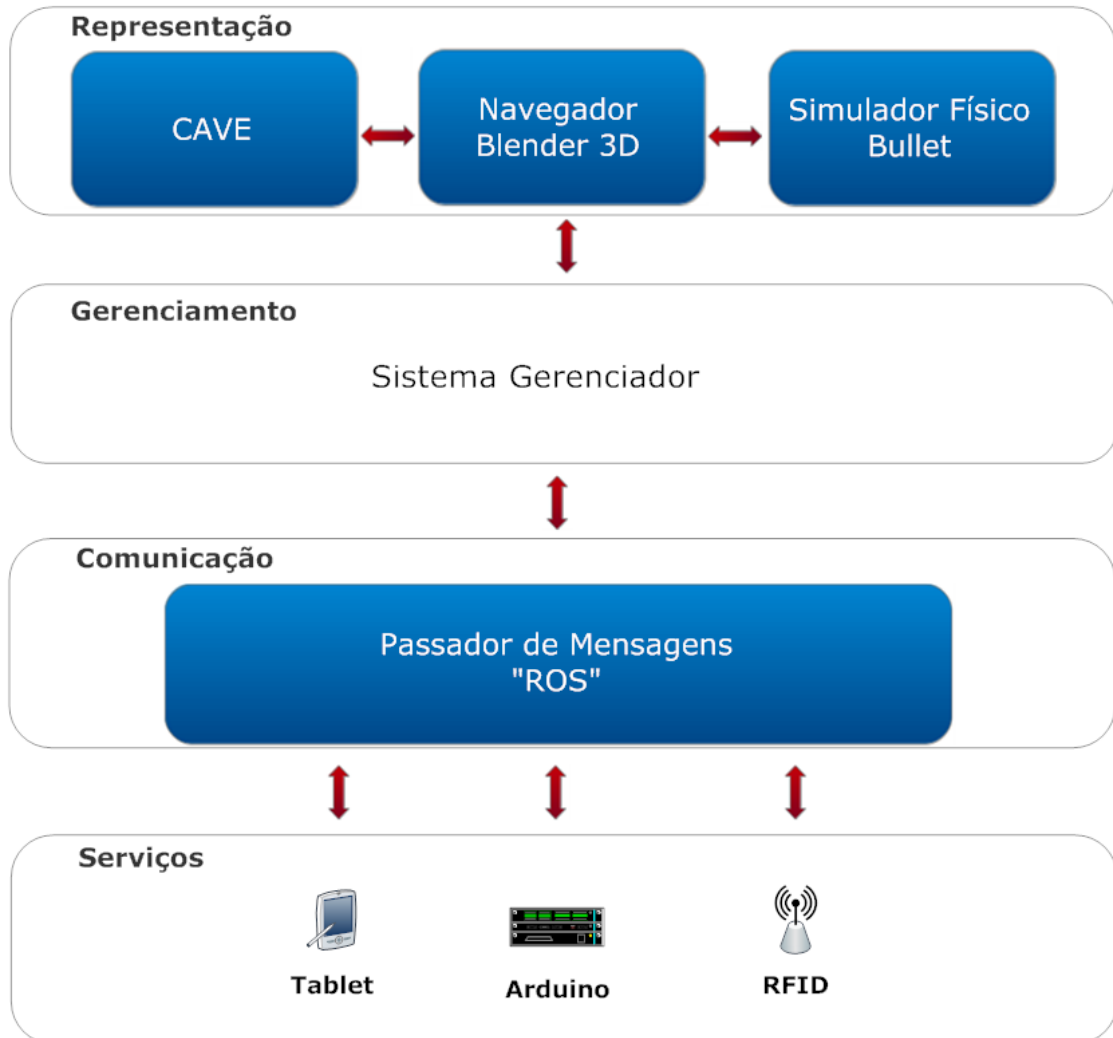


Figura 4.2: Representação da arquitetura do FAPS-Naval.

proposta é representada na Figura 4.2.

Nesta arquitetura pode-se identificar os seguintes componentes e serviços:

- nodo sensor RFID: Trabalhou-se com dois tipos de componentes RFID. O primeiro, é o leitor Acura RFID L-A201, de dimensões reduzidas e frequência de leitura 433MHz, que possui alcance de leitura máximo de 30 metros em campo aberto. Este leitor trabalha com *tags* ativas, visto a frequência de comunicação e a distância que o sinal deve percorrer, fazendo com que a *tag* possua uma fonte de alimentação para fornecer sinal. Já o segundo tipo de leitor, usado para simulações em tamanho reduzido, é o leitor Acura RFID AP-16, geralmente instalado em portas e liberações de acesso, que possui frequência 125kHz e distância de leitura máxima de aproximadamente

10 centímetros. O AP-16 utilizada *tags* passivas, no formato de cartões ou chaveiros de mesma frequência que o leitor.

- nodo sensor humidade: O sensor de umidade, desenvolvido utilizando o Arduino, que é uma plataforma de prototipagem eletrônica de *hardware* livre, tem por objetivo controlar índices de umidade relativa, seja do solo ou do ar. Este pode ser utilizado para controles meteorológicos em estaleiros, como por exemplo na recente base naval de construção de submarinos nucleares em Itaguaí/RJ. Uma aplicação do ROS pode capturar estes valores de leitura e armazená-los na base de dados para realizar controles necessários.
- nodo sensor temperatura: Muito semelhante ao sensor de humidade, também utiliza a plataforma Arduino e consegue detectar índices de temperatura, seja do ar, água ou de algum equipamento específico. Com este equipamento, é possível o controle de aquecimento de máquinas e motores, além da detecção de problemas na refrigeração de ambientes ou equipamentos.
- nodo localizador: O nodo localizador pode ser concebido utilizando um sensor Arduino chamado ArduIMU V3, que é uma Unidade de Medidas Inerciais (*Inertial Measure Unit* - IMU) que executa um sistema de controle de atitude (*Attitude Heading Reference System* - AHRS). Consiste em um *hardware* com três acelerômetros de eixo e três sensores de giro, além de suporte GPS e LEDs de estados. Esta plataforma pode ser utilizada para localização em tempo-real, ajuda inclusive em sistemas de *procurement* e suplementos. Identificação e previsão de atrasos pode trazer um grande ganho em performance nos processos de produção naval. Estaleiros como o EISA, do Rio de Janeiro, sofrem com atrasos de suprimentos e poderiam se beneficiar de um sistema que conseguisse prever estes problemas.

A camada de comunicação, implementada através do *middleware* ROS, realiza a convergência das informações adquiridas pelos nodos. Esta camada pode manipular uma grande quantidade de nodos, os quais adquirem e tratam informações em tempo real. Definidos quais são os tipos de informação que o nodo pode captar e transmitir ao sistema, estes dados são publicados no tópico do sistema gerenciador, que constantemente recebe e armazena os dados lidos.



Figura 4.3: Visão aérea do Estaleiro Rio Grande.

Na camada de representação, uma aplicação mostra a representação 3D dos diferentes componentes, campos de informações e ajuda, manuais, e relatórios, descritos através do ambiente de desenvolvimento Blender. Nela, é possível posicionar elementos e visualizar em tempo real onde se encontram objetos, equipamentos e equipes de trabalho desejados, tendo uma visão virtual do que acontece no funcionamento do estaleiro.

O sistema de gerenciamento foi implementado visando o tratamento das informações dos nodos e sua representação, visando a disponibilização dos serviços e componentes presentes no ciclo de vida do navio. A seguir descreve-se os serviços implementados a partir de atividades relacionadas ao BoL.

O FAPS-Navio aplicado ao BoL no Estaleiro Rio Grande

O FAPS-Navio foi aplicado em um estudo de caso no estaleiro Rio Grande. O estaleiro Rio Grande é um estaleiro recentemente construído e em atividade na cidade de Rio Grande, Rio Grande do Sul, Brasil. Atualmente foca em navios de apoio marítimo, embarcações de grande porte e plataformas, estando em vista e em andamento a construção de oito cascos para a Petrobras do tipo FPSO (navio-plataforma), estruturas que tiveram a produção incentivada pela exploração da camada pré-sal brasileira. Existem investimentos de aproximadamente R\$ 300 milhões previstos para completar a capacidade construtiva, pretendendo futuramente produzir também navios sonda de perfuração.

O ERG tem seu canteiro de obras apresentado na figura 4.3. O cenário escolhido estabelece-se na fase de montagem ou edificação, onde diferenças entre projeto e execução podem causar retrabalho ou problemas estruturais futuros. A verificação entre os dois é, na maioria das situações, realizada por encarregados da montagem e pelo controle de qualidade, que têm a responsabilidade de analisar e constatar se a obra pode prosseguir. Este trabalho é manual, usando folhas tarefa que descrevem a análise desejada, onde o funcionário deve assinar assim que concluir que o andamento está de acordo com as especificações. O processo se torna dispendioso, confuso e demorado, devido a extravios de documentação e inserções em planilhas de controle.

É verificada a utilização do FAPS-Navio no rastreamento, monitoramento e supervisão de uma série de chapas de aço que devem ser soldadas de acordo com a especificação em projeto. Abaixo, as diferentes visões e detalhamentos do processo para alguns dos possíveis *players* na tarefa:

Setor de projeto: Libera a realização da tarefa no sistema PSS. Ação serve de entrada ao setor de almoxarife.

1. Setor de almoxarife: Automaticamente visualiza a liberação, e depois de verificado o estoque, libera a realização da tarefa. O sistema aloca uma equipe de solda disponível.
2. Equipe de solda: O encarregado visualiza no sistema uma nova tarefa e o *framework*, através de registros passados por dispositivos RFID que cada funcionário e equipamento possuem, mostra quais estações de solda e equipamentos estão disponíveis.
3. Controle de Qualidade: Terminada a tarefa de soldagem, o encarregado utiliza o sistema para marcá-la como terminada. O controle de qualidade então automaticamente é notificado que pode fazer a verificação.
4. Administração: Depois que o controle de qualidade aprova a tarefa, todas as “assinaturas” virtuais necessárias são introduzidas no sistema e a tarefa é dada como encerrada. Devido aos dispositivos, como RFIDs, que a equipe utiliza, a administração tem conhecimento de quem está trabalhando onde e quando, além da porcentagem de conclusão das tarefas desempenhadas.

Os nodos Neste cenário os nodos utilizados são os sensores de RFID e tablets. Os sensores RFID estão conformados nas seguintes redes:

- Rede RFID para Rastreo de Funcionários: esta rede é formada por sensores RFID de longa distância, com leitura de 30 metros, para casos de detecção de presença em ambientes, e de leitores com baixa distância, para uso em portas de acesso, registrando entradas e saídas que permitem o rastreo dos funcionários no pátio e oficina. A informação de presença transmitida pelas tags é captada pelo ROS e repassada através de um tópico do sistema gerenciador, que constantemente lê e armazena os dados recebidos. Assim, todas as leituras de presença de *tags* realizadas pelos sensores são armazenadas no banco de dados. Um nodo do ROS controla todos os dispositivos RFID ligados na rede e responsabiliza-se em publicar as informações de presença, entrada ou saída no tópico do sistema gerenciador. Isso pode ser usado para rastrear pessoas dentro do ambiente de estaleiro, tornando possível inclusive controle de horas de trabalho, ou então, conhecimento de quais oficinas estão disponíveis em determinado momento.
- Rede RFID para Identificação de Paletes: Com identificação por radiofrequência em paletes de chapas armazenadas no pátio, é possível que um operador de guindaste ou caminhão rapidamente identifique qual grupo de chapas foi encarregado de transportar, podendo identificar também, por um sistema integrado, onde a carga deve ser depositada. Cada caminhão ou guindaste pode ter um sensor de longo alcance, funcionando em 12V de alimentação, comunicando por transmissão *wireless* ou 3G com o sistema de controle.
- Rede RFID de Rastreo para Máquinas de Solda: Controla o uso, a estação de trabalho e o funcionário que está utilizando a máquina de solda, prevendo manutenções preventivas baseadas em tempo de uso e determinação de responsáveis em caso de danos, mau uso ou extravio de equipamento. Uma *tag* de leitura acoplada no equipamento e os leitores presentes nas estações de solda fariam a detecção, tanto da etiqueta do equipamento, quanto da etiqueta do soldador, fazendo com que o sistema tenha conhecimento de quem está manipulando o equipamento em determinado momento.

- Acesso por *Tablets*: *Tablets* de tela com tamanho aproximado de 10.1 polegadas conseguem ter suporte e bom uso da interface *web* desenvolvida na camada de representação do FAPS. Dispositivos menores também podem ser utilizados, mas com uma interface redesenhada para dispositivos de menor porte. Com a possibilidade de uso deste recurso, encarregados e supervisores podem checar quais tarefas em andamento estão corretas e quais estão atrasadas, além de ter um controle geral sobre o seu setor. Neste caso, a interface *web* funciona como um nodo do sistema gerenciador, visto que ela não funciona apenas como um sistema de visualização dos dados, mas também como um meio de passar informações ao sistema, fazendo mudanças em processos e fornecendo instruções a equipes.

Como percebe-se, o FAPS-Navio permite o uso de diferentes tipos de *tags* RFID, bem como a comunicação entre diferentes dispositivos móveis e *tablets*.

Camada de comunicação A detecção de entidades e disponibilização de serviços é realizada pela camada de comunicação. É esta camada que permite com que dispositivos consigam armazenar suas informações na base de dados do FAPS. A camada de representação, com base nas informações adquiridas e nos serviços disponibilizados que estão registrados no banco de dados, permite a emissão de relatórios fidedignos e precisos, que auxiliam no cálculo de homem-hora das tarefas, controle de avanço da edificação, análise de problemas ou defeitos e etc. A camada de comunicação gerencia os diferentes dispositivos presentes no estaleiro, como RFIDs e sensores, fazendo com que as informações fornecidas por esses, que não são idênticas em formato de transferência de dados ou em tipo, consigam ser entendidas e armazenadas pela camada de gerenciamento.

Camada de representação O FAPS-Navio aplicado ao estaleiro Rio Grande possui uma camada de representação implementada através de um dispositivo de visualização avançada do tipo *V-CAVE* multi-projeção, bem como visualização 3D em estação de trabalho e *tablets*. Um modelo do ERG em 3D foi desenvolvido (ver figura 3.6), apresentando suas características estruturais e físicas. O modelo apresenta textura e iluminação capazes de fornecer impressões visuais fidedignas. A maquete tem seus elementos físicos simulados, permitindo a operação e simulação de guindastes e veículos de transporte, em um

sistema chamado Simulador Físico Bullet (Figura 3.2).

O sistema de representação apresenta, online, as atualizações dos componentes 3D associados aos serviços. O usuário pode navegar e interagir através do ambiente, visualizando a localização e estado dos insumos (funcionários e equipamentos).

Camada de gerenciamento Nesta implementação a camada de gerenciamento controla a emissão de tópicos para cada tipo de nodo. O nodo RFID, por exemplo, possui um controlador na camada de comunicação que recebe entradas de todos os RFIDs presentes na planta. Este controlador, passa para o gerenciador e este, com base nos dados que possui na base de dados, consegue armazená-los adequadamente ou disparar eventos programados. As informações vindas dos diversos serviços desempenhados pelos dispositivos do estaleiro chegam até esta camada através de tópicos. Existe um tópico para cada tipo de dispositivo, então quando o sistema monitora constantemente as entradas do tópico de RFID, que são postados em uma frequência de 10Hz, ele sabe a maneira adequada de tratar aqueles dados e armazená-los de maneira correta.

4.2.2 Considerações Finais

A arquitetura FAPS foi customizada para uso no ciclo de vida de navios. Este produto foi escolhido devido a complexidade de sua cadeia, bem como a importância deste segmento para a economia nacional e local. O ciclo de vida de um navio foi analisado à luz do paradigma CoPS/PSS. As fases e componentes do ciclo de vida foram identificadas, de forma a propiciar a instanciação do FAPS ao monitoramento, supervisão, controle e visualização das atividades e processos envolvidos. O FAPS-Navio foi implementado e aplicado em um estudo de caso no ERG. Alguns serviços foram implementados, associados ao rastreamento e monitoramento, bem como a visualização avançada do processo.

É interessante observar que o conceito e implementação se aplicam a inúmeras outras atividades. Por exemplo, na realização de inventário do estoque de peças e insumos, o trabalho manual é dispendioso e demorado, além de estar sujeito a erros. Utilizando identificação RFID e leitores o controle do estoque é automático, sem necessidade de interação humana. Isso auxilia até mesmo no planejamento de compras realizado pelo estaleiro.

4.3 Estudo de Caso - Estaleiro Rio Grande

O FAPS, devido a facilidade em comunicar dispositivos de diferentes naturezas, em teoria pode ser aplicado a qualquer indústria que necessite de um *framework* que auxilie o gerenciamento de tarefas e dispositivos. Ele desde o início foi pensado com o foco na indústria naval, devido ao cenário promissor da área no país, principalmente no sul do Brasil. A Universidade Federal do Rio Grande - FURG - é localizada na cidade de Rio Grande, onde recentemente instalou-se um dos maiores *clusters* navais do Brasil, em fase de construção e desenvolvimento. O Estaleiro Rio Grande, primeiro estaleiro que se instalou na região, serviu como caso base da aplicação do *framework* FAPS, sendo as metodologias de trabalho e necessidades de melhorias tecnológicas encontradas no ERG a inspiração para realizar este trabalho.

Desde o início das pesquisas, foram encontrados uma série de práticas nos estaleiros nacionais que, sob o ponto de vista da tecnologia da informação e da computação, não usufruem dos benefícios que novas práticas, pesquisas e tecnologias podem fornecer. Alguns dos principais pontos foram:

- Especificações de solda, peças e chapas passadas em folhas de papel;
- Dificuldade em localizar peças, chapas e equipamentos localizados dentro do canteiro de obras;
- Equipamentos de Proteção Individual (EPI) muitas vezes negligenciados em visitantes no estaleiro;
- Falta de controle sob qual atividade está sendo desempenhada por cada funcionário.

Na busca de fornecer uma possível solução para parte destes problemas, resolveu-se aplicar o *framework* FAPS neste ambiente com o uso de dispositivos RFID. Com sensores e etiquetas de identificação, problemas de localização, presença e proximidade podem ser resolvidos dentro de um ambiente controlado, fazendo uso de etiquetas de frequência adequada e sensores com alcance de leitura apropriados. Com a autorização de Engenheiros do ERG, foram permitidos testes (Figura 4.4) de dispositivos de leitura RFID, onde analisou-se os alcances de leitura para certas etiquetas, assim como a interferência de objetos, chapas metálicas e ambientes de solda.



Figura 4.4: Testes de dispositivos RFID realizados no Estaleiro Rio Grande.

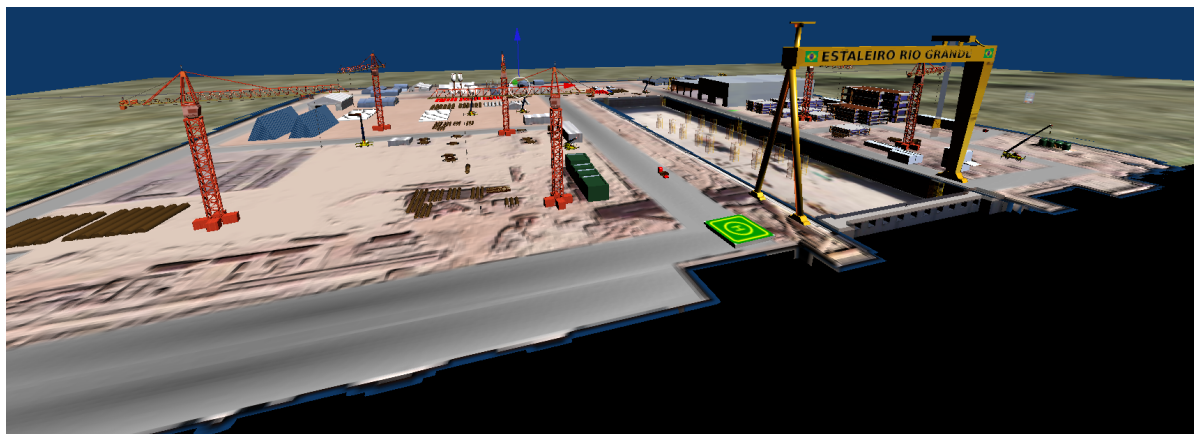


Figura 4.5: Maquete virtual utilizada nos testes do FAPS.

O estaleiro Rio Grande pode ser visualizado virtualmente na figura 4.5, que representa uma maquete virtual desenvolvida para simular as atividades que acontecem dentro do chão de fábrica. Os testes realizados demonstram dados coletados por dispositivos RFID, que através da camada de comunicação do FAPS passam seus dados para o sistema gerenciador, onde são armazenados na base de dados. Com as informações presentes no banco de dados, é então possível que uma aplicação da camada de representação, como a maquete virtual com interações por *scripts* Python, adquira estes dados e utilize-os para determinado propósito. No caso da animação, estes dados posicionam elementos virtuais, que representam os componentes reais, fazendo com que objetos reais em um ambiente sejam facilmente localizados dentro do ambiente virtual do estaleiro.

Os testes foram realizados em uma maquete real, que representa o Estaleiro Rio Grande. Nesta maquete, foram colocados dois leitores RFID para simular dois possíveis ambientes dentro do estaleiro. Na simulação, quando uma *tag*, que representa um equipamento, pessoa ou peça, se aproxima de um destes ambientes, o sensor RFID identifica a sua presença e passa para a aplicação de gerenciamento do FAPS, que por sua vez, permite que uma representação virtual do componente seja mostrada na renderização 3D. Além da maquete virtual mostrada anteriormente, uma aplicação *web* também pode mostrar dados de maneira a facilitar a leitura de informações e estados dentro do ambiente. A figura 4.6 mostra um exemplo de interface de visualização para localizar dados de componentes dentro de um estaleiro.

Dados Cadastrais			
Tag		0003602867	
Marca		Motor	
Número de Registro		123456789	

Suprimentos	Entrada	31/05/2012	15:56:33
Suprimentos	Entrada	23/03/2012	21:19:04
Montagem	Saida	23/03/2012	21:19:04
Montagem	Entrada	23/03/2012	21:05:31
Suprimentos	Saida	23/03/2012	21:05:31
Suprimentos	Entrada	23/03/2012	20:50:01
Suprimentos	Saida	23/03/2012	20:49:55
Suprimentos	Entrada	23/03/2012	20:43:39
Suprimentos	Saida	23/03/2012	20:42:19
Suprimentos	Entrada	23/03/2012	20:42:16

Figura 4.6: Interface *web* de uma aplicação de representação no FAPS.



Figura 4.7: Aplicação FAPS simulando interações de ambiente real em ambiente virtual.

Capítulo 5

Conclusão

Atualmente, há uma grande demanda no mercado pelo desenvolvimento de ferramentas de acompanhamento para projetos de estruturas de engenharia complexas, que consigam agilizar processos nos setores de *procurement*, facilitar o controle e a gerência do que acontece ao longo do ciclo de vida de produtos, objetivando aumentar a produtividade e a competitividade. Indústrias estrangeiras levam a maior parte das encomendas nacionais e o momento é propício para mostrar os benefícios de produzir em território nacional.

O trabalho de mestrado aqui apresentado teve por objetivo criar uma ferramenta para o monitoramento, supervisão e controle do ciclo de vida de grandes estruturas de engenharia, mais especificamente para a indústria naval, com problemas de engenharia reais, usando da literatura de diversas áreas de conhecimento e estudando as principais práticas de estado da arte encontradas na literatura atualmente. O objetivo do *Framework* de Acompanhamento Produto-Serviço (FAPS) é fornecer uma estrutura de trabalho que permita que indústrias consigam controlar melhor seus processos e incorporar seus equipamentos e sistemas em uma única solução integrada.

O FAPS é baseado em novas estratégias de produção, tais como CoPS/PSS, bem como de novos paradigmas e tecnologias industriais como IoT e Realidade Virtual. Com base na complexidade e expectativas regionais e nacionais de competitividade da indústria naval e *offshore*, o *framework* foi aplicado, testado e analisado em uma aplicação no ciclo de vida de navios. Um estudo de caso foi projetado junto ao Estaleiro Rio Grande.

Ao longo do trabalho, foi realizada uma revisão bibliográfica sobre as principais teorias e paradigmas utilizados, tanto em estratégias de produção, bem como em técnicas avança-

das de Internet das Coisas e conceitos de Arquitetura Orientada a Serviços, apresentadas no capítulo 2.

Com base na revisão bibliográfica, o capítulo 3 apresentou a estrutura e arquitetura do *framework* desenvolvido. Este é composto por 4 camadas principais. A camada de serviços representa os diversos dispositivos e seus respectivos serviços prestados e recebidos ao longo do seu funcionamento. A segunda camada, de comunicação, permite que seja possível fazer a comunicação entre dispositivos de natureza diferente trocarem informações, fornecendo suas informações também ao componente da terceira camada, de gerenciamento, onde estes dados são tratados adequadamente e armazenados de forma a serem utilizados por dispositivos e por sistemas externos que necessitam desta base de dados para cumprir alguma funcionalidade. Estes sistemas externos pertencem a camada de representação, que engloba as diversas aplicações que utilizam os dados e informações, fornecidos pela camada de serviços, seja para melhor representá-los, filtrá-los ou mesmo visualizá-los.

O capítulo 4 apresentou uma instanciação da proposta, integrando informações ao longo do ciclo de vida de um navio. A indústria de construção naval e *offshore* foi analisada, bem como o ciclo de vida de navios, sobretudo no cenário de retomada da construção naval brasileira. O estudo resultou no FAPS-Navio, que permitiu a aplicação dos conceitos trabalhados no início do ciclo de vida de um navio em construção no ERG.

Foram analisados os possíveis componentes e serviços existentes ao longo do ciclo de vida de um navio, identificando suas informações circulantes. Para a implementação da instanciação FAPS-Navio, foram desenvolvidas as camadas pertencentes a arquitetura FAPS, onde a camada de serviços abordou o uso de tecnologias de Identificação por Radiofrequência (RFID) e Sensores com base na plataforma Arduino para fornecer solução a problemas conhecidos e existentes nos ambientes de estaleiros nacionais. Ocorrendo a comunicação através da camada de comunicação baseada em ROS, a camada de gerenciamento permite com que aplicações de realidade virtual, usando animações virtuais de ambientes de estaleiros, permitam melhor entendimento, forneçam navegabilidade e melhor entendimento do estado do funcionamento, em tempo real e de histórico, em um estaleiro.

A aplicação do FAPS no estudo de caso, permitiu integrar diferentes serviços presen-

tes no início do ciclo de vida. Vários tipos de nodos permitiram a aquisição online das informações circulantes. A camada de comunicação permitiu a integração de diferentes dispositivos e a comunicação de dados. O gerenciador apresentou um conjunto de características necessárias ao acompanhamento e gerenciamento dos serviços. Uma representação 3D avançada permitiu a visualização, interação e cooperação entre os diferentes atores dos diferentes serviços e componentes do sistema. Com a aplicação do FAPS na área naval, foi possível alcançar o objetivo de gerar uma solução integrada (PSS), em problemas complexos que muitas vezes possuem atuação direta do cliente interessado (CoPS), onde a comunicação entre a rede de dispositivos heterogêneos é fundamental (IoT) e há uma enorme quantidade de serviços prestados e recebidos por diversas entidades atuantes nos processos (SOA).

5.1 Trabalhos Futuros

Como trabalhos futuros vislumbra-se o desenvolvimento de outros nodos de aquisição, tais como dispositivos de elementos *bluetooth*, que venham a fazer parte do sistema. Também seria interessante a adoção de padrões e protocolos industriais de comunicação, bem como dotar o sistema de inteligência, fazendo que autonomamente novos dispositivos possam ser detectados e integrados na rede de serviços. Finalmente, realizar mais testes e aplicações junto ao ciclo de vida de navios, verificando a integração de serviços e produtos ao longo do projeto, comissionamento e manutenção também são trabalhos futuros que serão realizados.

Bibliografia

- [Araujo et al., 2010] Araujo, F. O., Dalcol, P. R. T., and Longo, W. P. (2010). Proposta de metodologia para análise de sistemas setoriais de inovação: Aplicações na indústria de construção naval. *23º Congresso Nacional de Transporte Aquaviário, Construção Naval e Offshore - SOBENA*.
- [Atzori et al., 2010] Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Comput. Netw.*, 54(15):2787–2805.
- [Aurich et al., 2006] Aurich, J., Fuchs, C., and Wagenknecht, C. (2006). Life cycle oriented design of technical product-service systems. *Journal of Cleaner Production*, 14(17):1480 – 1494. Product Service Systems: reviewing achievements and refining the research agenda.
- [Baines et al., 2007] Baines, T. S., Lightfoot, H. W., Evans, S., Neely, A., Greenough, R., Peppard, J., Roy, R., Shehab, E., Braganza, A., Tiwari, A., Alcock, J. R., Angus, J. P., Bastl, M., Cousens, A., Irving, P., Johnson, M., Kingston, J., Lockett, H., Martinez, V., and Michele, P. (2007). State-of-the-art in product-service systems. *Proceedings of the Institution of Mechanical Engineers – Part B – Engineering Manufacture*, 221(10):1543–1552.
- [Blender, 2003] Blender, F. (2003). What is blender? *Blender Documentation*. <http://download.blender.org/documentation/html/x57.html> - Acessado em Abril de 2012.
- [Brax and Jonsson, 2009] Brax, S. A. and Jonsson, K. (2009). Developing integrated solution offerings for remote diagnostics - a comparative case study of two manufacturers. *International Journal of Operations and Production Management*, 29(5):539–560.

- [Carvalho et al., 2011] Carvalho, J., Penna dos Santos, R., Silva da Costa Botelho, S., Filho, N., Oliveira, R., and Santos, E. (2011). Hyper-environments: A different way to think about iot. In *Internet of Things (iThings/CPSCoM), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*, pages 25 –32.
- [Cova and Salle, 2007] Cova, B. and Salle, R. (2007). Introduction to the imm special issue on 'project marketing and the marketing of solutions' a comprehensive approach to project marketing and the marketing of solutions. *Industrial Marketing Management*, 36(2):138 – 146. Project Marketing and the Marketing of Solutions.
- [Davies and Brady, 2000] Davies, A. and Brady, T. (2000). Organisational capabilities and learning in complex product systems: towards repeatable solutions. *Research Policy*, 29(7-8):931–953.
- [Endrei et al., 2004] Endrei, M., Ang, J., Arsanjani, A., Chua, S., Comte, P., Krogdahl, P., Luo, M., and Newling, T. (2004). *Patterns - Service-Oriented Architecture and Web Services*. IBM Redbooks.
- [Favarin, 2008] Favarin, J. (2008). Estratégia para a navepeças brasileira: uma abordagem por competências. *Centro de Estudos em Gestão Naval da Escola Politécnica da USP*.
- [Gebauer, 2008] Gebauer, H. (2008). Identifying service strategies in product manufacturing companies by exploring environment-strategy configurations. *Industrial Marketing Management*, 37(3):278 – 291. The Transition from Product to Service in Business Markets.
- [Goetz, 2009] Goetz, S. D. (2009). Bpm cbok leitfaden für das prozessmanagement herausgegeben von der eabpm (european association of business process management). *Business Process Management Common Body of Knowledge*.
- [Hammer and Champy, 2003] Hammer, M. and Champy, J. (2003). *Reengineering the Corporation: A Manifesto for Business Revolution*. HarperBusiness Essentials Series. HarperBusiness Essentials.

- [Heighes, 1997] Heighes, T. (1997). Quantitative Indication for complex product systems and their value to the UK economy. In *7th International Forum on Technology Management*, pages 3–7, Kyoto, Japan.
- [Hobday and Rush, 2001] Hobday, M. and Rush, H. (2001). Technology management in complex product systems (COPS) - ten questions answered. *International Journal of Technology Management*, 17:618–638.
- [Hobday et al., 2000] Hobday, M., Rush, H., and Tidd, J. (2000). Innovation in complex products and systems. *Research Policy*, 29(7-8):793 – 804.
- [Huang and Li, 2010] Huang, Y. and Li, G. (2010). Descriptive models for internet of things. In *Intelligent Control and Information Processing (ICICIP), 2010 International Conference on*, pages 483 –486.
- [IBM, 2008] IBM (2008). Product lifecycle management for shipbuilding and offshore. *IBM Product Lifecycle Management*, page 16.
- [Ibrahim, 2009] Ibrahim, N. (2009). Orthogonal classification of middleware technologies. In *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2009. UBI-COMM '09. Third International Conference on*, pages 46 –51.
- [Jia and Jinke, 2011] Jia, C. and Jinke, Y. (2011). Research on the informatization of shipbuilding industry based on the investigation on Wuchang shipyard. *Proceedings of the 7th International Conference on Innovation and Management*, pages 1532–1536.
- [Josuttis, 2007] Josuttis, N. (2007). *SOA in Practice: The Art of Distributed System Design*. Theory in practice. O'Reilly.
- [jQuery Foundation, 2012] jQuery Foundation, T. (2012). jquery. <http://www.jquery.com/> - Acessado em 03/02/2012.
- [Koshizuka and Sakamura, 2010] Koshizuka, N. and Sakamura, K. (2010). Ubiquitous ID: Standards for ubiquitous computing and the internet of things. *Pervasive Computing, IEEE*, 9(4):98 –101.

- [Kranz et al., 2011] Kranz, M., Möller, A., and Roalter, L. (2011). Robots, objects, humans: Towards seamless interaction in intelligent environments. In *1st International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS 2011)*, Algarve, Portugal.
- [Lifton, 2007] Lifton, J. (2007). *Dual Reality - An Emerging Medium*. PhD thesis, Massachusetts Institute of Technology - Department of Media Arts and Sciences.
- [Lifton and Paradiso, 2009] Lifton, J. and Paradiso, J. A. (2009). Dual reality - merging the real and virtual.
- [Lima and Velasco, 1998] Lima, E. T. and Velasco, L. (1998). Construção naval no brasil: Existem perspectivas? *Revista do BNDES*, 5(10):167–194.
- [MacKenzie et al., 2006] MacKenzie, C. M., Laskey, K., McCabe, F., Brown, P. F., and Metz, R. (2006). Reference model for service oriented architecture 1.0. *OASIS Standard*.
- [Mein, 2004] Mein, K. (2004). Getting down and dirty with the blender source. *Blender Conference*. <http://www.blender.org/community/blender-conference/blender-conference-2004/proceedings/kent-mein/full-paper/> - Acessado em Abril de 2012.
- [Mont, 2002] Mont, O. (2002). Clarifying the concept of product-service system. *Journal of Cleaner Production*, 10(3):237 – 245.
- [Paradiso and Landay, 2009] Paradiso, J. and Landay, J. (2009). Guest editors' introduction: Cross-reality environments. *Pervasive Computing, IEEE*, 8(3):14–15.
- [Passos, 2007] Passos, P. S. O. (2007). Visão atual e perspectivas do programa de expansão e modernização da marinha mercante. *FENASHORE*.
- [Pezzotta, 2010] Pezzotta, G. (2010). Integrated engineering of product-services throughout the life-cycle phases. In *Research Report*. CELS - Research Center on Logistics and After Sales Service, University of Bergamo.
- [Pham, 2010] Pham, T. (2010). Model-view-controller design. *CSS 555 - Evaluating Software Design*.

- [PHP-Group, 2001] PHP-Group, T. (2001). What is php? *PHP FAQ*. <http://www.php.net/manual/en/faq.general.php#faq.general.what> - Acessado em 01/02/2012.
- [Roalter et al., 2011] Roalter, L., Moller, A., Diewald, S., and Kranz, M. (2011). Developing intelligent environments: A development tool chain for creation, testing and simulation of smart and intelligent environments. In *Intelligent Environments (IE), 2011 7th International Conference on*, pages 214–221.
- [ROS, 2012] ROS (2012). Ros wiki. *ROS Documentation*. <http://www.ros.org/wiki/ROS> - Acessado em 23 de Maio de 2012.
- [Rummler and Brache, 1995] Rummler, G. and Brache, A. (1995). *Improving performance: how to manage the white space on the organization chart*. JOSSEY BASS BUSINESS AND MANAGEMENT SERIES. Jossey-Bass.
- [Schilit et al., 1994] Schilit, B., Adams, N., and Want, R. (1994). Context-aware computing applications. In *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications, WMCSA '94*, pages 85–90, Washington, DC, USA. IEEE Computer Society.
- [SOA4All, 2010] SOA4All (2010). Large-scale integrating project. <http://www.soa4all.eu/project.html>. Funded by the European Seventh Framework Programme. Reference: 215219.
- [Tukker, 2004] Tukker, A. (2004). Eight types of product-service system - eight ways to sustainability? experiences from suspronet. *Business Strategy and the Environment*, 13(4):246–260.
- [uID Center, 2011] uID Center (2011). <http://www.uidcenter.org/>. Acessado em Maio de 2012.
- [W3C, 2008] W3C (2008). Extensible markup language (xml) 1.0 (fifth edition). *W3C Recommendation*. <http://www.w3.org/TR/REC-xml/#sec-origin-goals> - Acessado em Maio de 2012.

- [Weiser, 1999] Weiser, M. (1999). The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11.
- [Wetzstein et al., 2009] Wetzstein, B., Leitner, P., Rosenberg, F., Brandic, I., Dustdar, S., and Leymann, F. (2009). Monitoring and analyzing influential factors of business process performance. In *Enterprise Distributed Object Computing Conference, 2009. EDOC '09. IEEE International*, pages 141 –150.
- [Ying-Tao and Khim-Teck, 2006] Ying-Tao, R. and Khim-Teck, Y. (2006). Research challenges on complex product systems (cops) innovation. *Journal of the Chinese Institute of Industrial Engineers*, 23(6):519–529.